B. E. SYLLABUS

COMPUTER SCIENCE & ENGINEERING

V & VI SEMESTER

With

Scheme of Teaching

& Examination

DEPARTMENT: COMPUTER SCIENCE & ENGINEERING

Sl. No.	Faculty Name	Qualification	Designation
1.	Dr. Niranjan.N. Chiplunkar	Ph.D	Principal
2.	Dr. K R Udaya Kumar Reddy	Ph.D	Professor & Head
3.	Dr. Udaya Kumar K Shenoy	Ph.D	Professor
4.	Dr. Deviprasad M	Ph.D	Professor
5.	Dr. D.K. Sreekantha	Ph.D	Professor
6.	Dr. Seetharam K	Ph.D	Professor
7.	Mrs. Jyothi Shetty	M.Tech, (Ph.D)	Asso. Prof
8.	Mrs. Sharada Udaya Shenoy	M.Tech, (Ph.D)	Asso. Prof
9.	Mr.Venugopala P.S.	M.Tech, (Ph.D)	Asso. Prof
10.	Mr. RoshanFernandes	M.Tech, (Ph.D)	Asso. Prof
11.	Mr. Radhakrishna D	M.Tech, (Ph.D)	Asso. Prof
12.	Mrs. Sarika Hegde	M.Tech, (Ph.D)	Asso. Prof
13.	Mr. Raju K	M.Tech, (Ph.D)	Asso. Prof
14.	Mr. Sudeepa K.B	M.E., (Ph.D)	Asso. Prof
15.	Mr. Pradeep Kanchan	M.Tech, (Ph.D)	Asst. Prof Gd III
16.	Mr. Ravi B	M.Tech, (Ph.D)	Asst. Prof Gd III
17.	Mr. VijayaMurari T	M.Tech, (Ph.D)	Asst. Prof Gd III
18.	Mrs. Pallavi KN	M.Tech(Ph.D)	Asst. Prof Gd II
19.	Mr. Ranjan Kumar HS	M.Tech	Asst. Prof Gd II
20.	Mrs. Anisha P Rodrigues	M.Tech, (Ph.D)	Asst. Prof Gd II
21.	Mr. Raghunandan KR	M.Tech	Asst. Prof Gd II
22.	Mrs. Minu P. Abraham	M.Tech	Asst. Prof Gd II
23.	Mr. Ramesha Shettigar	M.Tech	Asst. Prof Gd II
24.	Mr. SampathKini	M.Tech	Asst. Prof Gd II
25.	Mr. Mahesh Kini. M	M.Tech	Asst. Prof Gd II
26.	Mrs. Asmita Poojary	M.Tech	Asst. Prof Gd II
27.	Mrs. Shruthi M	M.Tech, (Ph.D)	Asst. Prof Gd II

28.	Ms. Savitha	M.Tech	Asst. Prof Gd I
29.	Mr. Sannidhan M.S	M.Tech	Asst. Prof Gd I
30.	Mr. SunilkumarAithal	M.Tech	Asst. Prof Gd I
31.	Mrs. Keerthana B. Chigateri	M.Tech	Asst. Prof Gd I
32.	Mr. Pawan Hegde	M.Tech	Asst. Prof Gd I
33.	Mrs. Shabari Shedthi. B	M.Tech, (Ph.D)	Asst. Prof Gd I
34.	Mr. Naveen Chandawerkar	M.Tech, (Ph.D)	Asst. Prof Gd I
35.	Mr. Krishna Prasad Rao	M.Tech	Asst. Prof Gd I
36.	Mr. Shashank Shetty	M.Tech	Asst. Prof Gd I
37.	Mrs. Divya Jennifer D'Souza	M.Tech	Asst. Prof Gd I
38.	Mr. Puneeth R.P	M.Tech	Asst. Prof Gd I
39.	Mrs. Shilpa Karegoudar	M.Tech	Asst. Prof Gd I
40.	Mrs. Rajalaxmi Prabhu	M.Tech	Asst. Prof Gd I
41.	Dr. Mohammed Javed	Ph.D	Asst. Prof Gd I
42.	Dr. Aravinda C V	Ph.D	Asst. Prof Gd III
43.	Mr. Sandeep Kumar Hegde	M.Tech	Asst. Prof Gd I
44.	Ms. Ankitha A Nayak	M.Tech	Asst. Prof Gd I
45.	Mrs. Swathi Pai M	M.Tech	Asst. Prof Gd I
46.	Ms. Rajashree	M.Tech	Asst. Prof Gd I

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION:

To be a center of excellence in Computer science & Engineering education and research, empower the lives of individuals to fulfill their academic excellence, professional passions, and partnership for community development..

MISSION :

- To impart both theoretical and practical knowledge through the state-of-the-art concepts and technologies in Computer Science and Engineering.
- To inculcate values of professional ethics, leadership qualities and lifelong learning.
- To create professionals for employment in industry, research, higher education, and entrepreneurship to benefit the society.

Programme Educational Objectives (PEOs):

1. Graduates will be capable of practicing principles of Computer Science & Engineering, Mathematics and Engineering sciences to solve problems that are appropriate to the discipline
2. Graduates will be able to contribute to their profession and society
3. Graduates will be employed in computing profession or engaged in learning to pursue higher education

Programme Outcomes (POs):

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

- Apply the knowledge of engineering science and mathematics in solving problems that are appropriate to the discipline
- Apply the knowledge of computing both hardware and software aspects to the solution of real-world engineering problems in the discipline
- Design & develop algorithms, programs, and projects using various software and modern tools appropriate to software industry or Research & Development activities in the discipline

Graduate Attributes :

Sl. No.	Graduate Attributes
a	Engineering Knowledge
b	Problem Analysis
c	Design / development of solutions
d	Conduct investigations of complex problems
e	Modern tool usage
f	The engineer and society
g	Environment and sustainability
h	Ethics
i	Individual and team work
j	Communication
k	Project management and finance
l	Life-long learning

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHEME OF TEACHING AND EXAMINATION

V SEMESTER B.E. **33 Hours / Week**

Sl. No.	Sub. Code	Subject	Theory/Tuto./Prac./Self Study	Total Hrs./Week	C.I.E	S.E.E	Credits
1	15CS501	Compiler Design	4+0+2+0	6	50+50	50+50	5
2	15CS502	Relational Database Management System	4+0+2+0	6	50+50	50+50	5
3	15CS503	Microprocessors & Peripherals	4+2+2+0	8	50+50	50+50	6
4	15CS504	Software Engineering	4+0+0+0	4	50	50	4
5	15CS505	Operating Systems	4+0+0+S	4	50	50	4
6	15CS51Y	Elective – I	3+0+0+0	4	50	50	3
7	15IL001	Employability Skill Development	0+1+0+0	1	50	0	0
		TOTAL	**33**	**33**	**550**	**450**	**27**

ELECTIVE – I

15CS511	Advanced UNIX Programming
15CS512	Signals and Systems theory
15CS514	Data Compression
15CS515	Operations Research
15CS516	Program Verification
15CS517	Introduction Internet of Things

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHEME OF TEACHING AND EXAMINATION

VI SEMESTER B.E. **32 Hours / Week**

Sl. No.	Sub. Code	Subject	Theory/Tuto./Prac./Self Study	Total Hrs./Week	C.I.E	S.E.E	Credits
1	15CS601	Computer Graphics & Multimedia	4+0+2+0	6	50+50	50+50	5
2	15CS602	Computer Networks	4+0+2+0	6	50+50	50+50	5
3	15CS603	Java & Internet Technologies	4+0+2+0	6	50+50	50+50	5
4	15CS604	Software Testing and Automation	3+1+0+S	4	50	50	3
5	15CS61Y	Elective – II	3+0+0+0	3	50	50	3
6	15CS62Y	Elective – III	3+0+0+0	3	50	50	3
7	15CS605	IGW*	0+3*+0+0	3	-	-	-
8	15IL002	Employability Skill Development	0+1+0+0	1	50	0	0
TOTAL			**32**	**32**	**500**	**450**	**24**

*** These classes will be held only during the first week of the semester.**

ELECTIVE – II

15CS611	Pattern Recognition
15CS612	System Simulation and Modeling
15CS613	Distributed Systems
15CS614	Advanced DBMS
15CS615	Cloud Computing and Infrastructure Management
15CS616	Managing Big data
15CS617	Information and Storage Management

ELECTIVE – III

15CS621	VLSI Design
15CS622	Multi core Architecture &Programming
15CS623	Microcontrollers
15CS624	Embedded and real time Systems
15CS625	Programming languages
15CS626	Multicast Communications
15CS627	Digital Signal Processing.

COMPILER DESIGN

Sub Code : 15CS501	**Credits : 04**
Hrs/Week : 4+0+2+0	**Total Hours : 52**

Course Learning Objectives:

1. Outline lexical analysis, use of regular expressions, transition diagrams and scanner-generator tools.

2. Get the idea of major parsing techniques top-down (recursive-descent, LL) and Bottom up parsers

3. Discuss LR parsers using items sets and parsing tables.

4. Make use of the principal ideas in syntax-directed definitions and syntax-directed translations, different code optimization techniques and code generation.

5. Describe how to construct of basic blocks, generation of code from expressions and basic blocks, and register-allocation techniques.

UNIT – I

Overview of the Translation Process, A Simple Compiler, Difference between interpreter, assembler and compiler. Overview and use of linker and loader, types of Compiler, The Phases of a Compiler

Lexical Analysis:

Lexical Analysis- The Role of Lexical Analyzer, Input Buffering, Specifications of Tokens, Recognition of Tokens, A Language for Specifying Lexical Analyzer, LEX programming.

Syntax Analysis: Context-free Grammars, ambiguity, associativity, precedence, Un ambigous grammars **12 Hours**

UNIT – II

Syntax Analysis:

The Role of the Parser, Top-down Parsing, Bottom-up Parsing, Operator-Precedence Parsing. **10 Hours**

UNIT – III

Syntax Analysis:

LR Parsers –Simple LR (SLR), Canonical LR (CLR), Look Ahead LR (LALR) Parsers, YACC programming. **10 Hours**

UNIT – IV

Syntax-Directed definitions, Constructions of Syntax Trees, Bottom-up Evaluation of S-attributed definitions, L-attributed definitions.

Intermediate Code Generation:

Intermediate Languages, Declarations, Assignments, Boolean Expressions **10 Hour**

UNIT – V

Code Generation:

Issues in the design of Code Generator, The Target Machine, Run-time Storage Management, Basic blocks and Flow graphs, Next-use information, A Simple Code Generator, Register Allocation and Assignment, The DAG representation of Basic Blocks

Code Optimization:
Introduction, The Principle of Optimization, Optimization of Basic Blocks, Loops in flow graphs. **10 Hours**

Course Outcomes

At the end of the course the student will be able to:

1. Outline different translators, learn applications of regular expression, write lexical analyzer using transition diagrams, understand the role of lexical analyzer in the first phase of compiler and understand other translators
2. Describe the role of syntax analyzer in a compiler, learn different top down and bottom up parsers and implement them and write YACC programs and implement LR parsers using YACC.
3. Construct LR parsing tables and learn applications of Context free grammars.
4. Make use of use of Syntax-Directed Translation in parsing and code generation, write Syntax-Directed Translation / Definition, for a given problem and generate intermediate code expressions, different language constructs.
5. Explain the use of Basic blocks and Flow graphs in code generation, the use of Syntax tree and DAG in code generation, working of simple code generator and write simple code generator for a subset of C language.

Mapping of POs & COs:

POs COs	a	b	c	d	E	f	g	h	i	j	k	L
1	H				H							
2	H	L			H							
3		M			H							
4		M			H							
5		M			H							H

H : High M: Medium L : Low

Text Book:

1. Alfred W Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman, Compilers- Principles, Techniques and Tools, Addison-Wesley, 2006.

REFERENCE BOOKS:

1. Andrew W Apple, Modern Compiler Implementation in C, Cambridge University Press, 1997.
2. Kenneth C Louden , Compiler Construction Principles & Practice, Thomson Education, 1997.
3. John R. Levine, Tony Mason, Doug Brown, LEX and YACC, O'Reilly Publication

****

COMPILER DESIGN LABORATORY

Sub Code : 15CS501 **Credits : 01**
Hrs/Week : 0+0+2+0

1. Implementation of lexical analyzer programs (lex programs).
2. Implementation of programs related to compilers (YACC programs)
3. Design and implementation of a mini project related to the area of compiler design. (Ex: Assemblers, lexical analyzer, any phase of compiler etc.)

****

RELATIONAL DATABASE MANAGEMENT SYSTEMS

Sub Code : 15CS502 **Credits : 04**
Hrs/Week : 4+0+0+0 **Total Hours : 52**

Course Learning Objectives:

This Course will enable students to

1. **Explain** the concept of databases, database management systems, database structures and how they work.
2. **Make use of** Entity-Relationship Modeling for creating simple databases from the real world scenarios.
3. **Write** structured query language (SQL) statements.
4. **Normalize** a database using Normalization Rules**.**
5. **Describe** database design concepts and algorithms.
6. **Discuss** the issues associated with Transaction Processing and Recovery.

UNIT – I

Introduction to database systems

Introduction, Characteristics of the Database approach, Actors on the scene, Advantages of using the DBMS approach, Data models, Schemes and Instances, Three Schema Architecture and Data Independence.

Entity-Relationship Model

Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the , ER Design for the COMPANY Database; ER Diagrams, Naming Conventions and Design Issues. **10 Hours**

UNIT – II

Relational Model And Relational Algebra

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations and Dealing with Constraint Violations;Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations : JOIN and DIVISION ;Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping. **11 Hours**

UNIT – III

SQL-The Relational Database Standard

SQL Data Definition and Data Types, Specifying Basic Constraints in SQL, Schema Change Statements in SQL; Basic Queries in SQL; More Complex SQL Queries; Insert, Delete and Update Statements in SQL; Additional Features of SQL; Views (Virtual Tables) in SQL; Database Programming: Issues and Techniques; Embedded SQL. **10 Hours**

UNIT – IV

Database Design

Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form; Properties of Relational Decompositions; Algorithms for Relational Database Schema Design; Multivalued Dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form. **10 Hours**

UNIT – V

Transaction Management

The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of Locking; Transaction Support in SQL ;Introduction to Crash Recovery; 2PL,Serializability and Recoverability; Introduction to Lock Management; Lock Conversions; Dealing with Deadlocks; Specialized Locking Techniques; Concurrency Control without Locking; Introduction to recovery: Recovery Concepts, Recovery Techniques Based on Deferred Update, Recovery Techniques Based on Immediate Update, Shadow Paging, The Aries Recovery Algorithm. **11 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. **Identify and Apply** the concept of databases, database management systems, ER modeling for designing simple databases.
2. **Summarize** the concepts related to relational model and **solve** database queries using relational algebra.
3. **Write** database queries using Structured Query Language (SQL).
4. **Design and develop** databases from the real world by applying the concepts of Normalization and design algorithms.
5. **Outline** the issues associated with Transaction Processing and Recovery.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1	**H**	**H**			**M**						**M**	
2		M			**H**						**M**	
3		**M**			**H**						**M**	
4	**H**	**H**	**M**								**H**	
5			**L**		**H**			**M**	**L**			

H : High M: Medium L : Low

TEXT BOOKS:

1. Elmasri and Navathe: *Fundamentals of Database Systems* (Fourth Edition), Pearson Education, 2003
2. Raghu Ramakrishnan and Johannes Gehrke : *Database Management Systems* (Third Edition), McGraw-Hill, 2003
3. Silberschatz, Korth and Sudharshan: *Database System Concepts*, Fourth Edition, Mc-GrawHill, 2002

****

RELATIONAL DATABASE MANAGEMENT SYSTEMS LABORATORY

Sub Code : 15CS502 **Credits : 01**
Hrs/Week : 0+0+2+0

1. Design and implementation of SQL queries involving various constructs of SQL as discussed in the Unit-III of the syllabus.
2. Implementation of a mini project that involves a user interface design, database design and design of SQL queries to suit the need of the designed application.

****

MICROPROCESSORS AND PERIPHERALS

Subject Code: 15CS503 **Credits: 4**
Hours/Week: 4+1+3+0 **Total Hours: 52**

Course Learning Objectives

This Course will enable students to:

1. **Outline** the internal architecture of 8086 microprocessor, concept of addressing modes, instruction set and develop and execute basic programs.
2. **Develop and Execute** modular assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
3. **Interface** microprocessor to external I/O devices namely logic controller, stepper motor, seven segment display, DAC, and keypad.
4. **Generate** machine code for 8086 instructions and **Outline** the working and benefits of 8259A Priority Interrupt Controller.
5. **Understand** the hardware components of 8086 microprocessor and compare the salient features of advanced microprocessors.

UNIT – I

8086 internal architecture: The Programming Model, Multipurpose registers, Special purpose Registers, Segment registers. Real mode memory addressing, Protected mode memory addressing, Flat mode addressing.

8086 Addressing modes: Register addressing, Immediate addressing, Direct addressing, Register Indirect addressing, Based addressing with displacement, Indexed addressing with displacement, Based Indexed addressing, Based Indexed addressing with displacement.

8086 Instruction Set - 1: Data transfer instructions (including I/O transfers), Binary arithmetic instructions, Decimal (BCD, ASCII) arithmetic instructions, Logical instructions, Shift and rotate instructions, Control transfer instructions.

8086 programming based on Instruction Set - 1: Programs based on data transfer instructions, binary arithmetic instructions, logical instructions, shift and rotate instructions, control transfer instructions. **12 Hours**

UNIT – II

Modular Programming: Using procedures, Using macros, Comparison between procedure and macro.

Data Conversions: ASCII to BCD/Hexadecimal, BCD/Hexadecimal to ASCII, 8 bit BCD to Hexadecimal, 8 bit Hexadecimal to BCD.

Using the Key Board and Video Display: DOS & BIOS interrupts, Disk files, Example Programs.

8086 Assembler Directives: ASSUME, DB, DW, DD, END, ENDP, ENDS, EQU, EVEN, EXTRN, GLOBAL, GROUP, INCLUDE, OFFSET, PROC, PUBLIC, SEGMENT, MACRO, ENDM. **10 Hours**

UNIT – III

8086 Instruction Set - 2: String instructions, Flag control instructions, Miscellaneous instructions. Programs based on String instructions.

Digital Interfacing: Methods of parallel data transfer, 8255A internal block diagram and system connections, 8255A operational modes and initialization, Constructing and sending 8255A control words. (Refer Text Book 2).

Interfacing 8086 microprocessor to Logic Controller Interface, Seven Segment Display Interface, Stepper Motor Interface, DAC Interface, and Keypad Interface. **10 Hours**

UNIT – IV

8086 Instruction Format (MOV instruction only): Generating machine code for register to register data transfer, memory/register to register/memory data transfer, immediate data transfer, segment register data transfer.

Combining Assembly Language with C/C++: Using Assembly Language with C/C++ for 16-Bit DOS Applications and 32-Bit Applications.

8086 Interrupts: 8086 Interrupts and Interrupt responses, 8086 Interrupt types, 8259A Priority Interrupt Controller – 8259A overview and system connections and cascading, Initializing an 8259A including Initialization Command Words (ICW) and Operational Command Words (OCW). (Refer Text Book 2). **10 Hours**

UNIT – V

8086 Hardware Specifications: 8086 pin functions, Bus buffering and latching (8086 only), Simplified 8086 write and read bus cycle, Minimum mode versus Maximum mode.

Introduction to Pentium Microprocessors: Pentium memory system, Branch prediction logic, Cache structure, Superscalar architecture, Special Pentium registers, Pentium memory management.

Comparison: Comparison of all Intel microprocessors in terms of clock frequency, Register size, Data bus size, maximum address space. (Refer Text Book 2, Page Number 16.13).

10 Hours

Text Books:

1. Barry B Brey: The Intel Microprocessors, 8th Edition, Pearson Education, 2009.
2. Douglas V Hall: Microprocessors and Interfacing, Tata McGraw-Hill Publication, Second Edition, 2006.

References:

1. Liu & Gibson: Microcomputer Systems: The 8086/8088 Family Architecture, Programming and Design, PHI, 2006.
2. Carl Hamachar, Zvonko Vranesic, Safwat Zaky: Computer Organization, Tata McGraw-Hill Publication, 2001.

MOOC:

NPTEL: http://nptel.ac.in/courses/108107029/39

Course Outcomes

At the end of the course the student will be able to:

- **Analyse** the internal architecture of 8086 microprocessor, concept of addressing modes and instruction assembling and to develop simple 8086 programs.
- **Develop and Execute** modular assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
- **Interface** microprocessor to external I/O devices namely logic controller, stepper motor, seven segment display, DAC, keypad, and elevator.
- **Understand** how to generate machine code for 8086 and understand the working and benefits of 8259A Priority Interrupt Controller.
- **Understand** the hardware components of 8086 and analyse the salient features of advanced microprocessors.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H	H										H
2	H				H							M
3		H			H					L		M
4		H			H					L	M	M
5		H			H					L		H

H: High M: Medium L: Low

****

MICROPROCESSORS AND PERIPHERALS LABORATORY

Sub Code : 15CS503 **Credits : 02**
Hrs/Week : 0+0+3+0

Course Learning Objectives:

This Course will enable students to

1. **Write and Execute** assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
2. **Interface** microprocessor to external I/O devices like logic controller, stepper motor, seven segment display, DAC, keypad, elevator.
3. **Implement** Real Mode programming in Pentium.

Students have to write, execute and test programs covering the syllabus of 13CS503.
Typical problems that may be tried are

1. Searching
2. Sorting
3. String manipulation
4. usage of Macros and subroutines
5. DOS interrupt usage
6. BIOS interrupt usage
7. Keyboard interface
8. Display interface
9. Logic controller interface
10. DAC interface
11. Stepper Motor control interface
12. Elevator interface
13. Some examples of Real Mode programming in Pentium.

Course Outcomes:

At the end of the course the student will be able to:

1. **Develop and Execute** assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
2. **Interface** microprocessor to external I/O devices like logic controller, stepper motor, seven segment display, DAC, keypad, elevator.
3. **Implement** Real Mode programming in Pentium.

Mapping of POs & COs:

P Os COs	a	b	c	d	e	F	g	h	i	j	k	L
1		H			M				L			M
2		H			H				L			M
3		H			H				L			M

H : High M: Medium L : Low

****

SOFTWARE ENGINEERING

Sub Code : 15CS504 **Credits : 04**

Hrs/ week : 4+0+0+0 **Total Hours : 52**

Course Learning Objectives:

On successful completion of this course, students will be able to:

1. **Define** software engineering, recognize its importance and code of conduct
2. **Explain** the phases and activities of the software development process
3. **Compare** several software engineering techniques.
4. **Oraganize** project, people, time and other resources
5. **Justify** the relavance of validation and verification, evaluation and costig in software engineering process.

UNIT – I

Overview

Introduction: FAQ's about software engineering, Professional and ethical responsibility. Software Processes: Models, Process iteration, Process activities; The Rational Unified Process; Computer-Aided Software Engineering.

Requirements

Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document. Requirements Engineering Processes: Feasibility studies; Requirements elicitation and analysis; Requirements validation; Requirements management. **10 Hours**

UNIT – II

System models, Project Management

System Models: Context models; Behavioral models; Data models; Object models; Structured methods.

Project Management: Management activities; Project planning; Project scheduling; Risk management. **10 Hours**

UNIT – III

Software Design

Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles.

Object-Oriented design: Objects and Object Classes; An Object-Oriented design process; Design evolution. **11 Hours**

UNIT – IV

Development

Rapid Software Development: Agile methods; Extreme programming; Rapid application development.

Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution. **11 Hours**

Verification and Validation

Verification and Validation: Planning; Software inspections; Automated static analysis; Verification and formal methods.

Software testing: System testing; Component testing; Test case design; Test automation.

UNIT - V

Management

Managing People: Selecting staff; Motivating people; Managing people; The People Capability Maturity Model.

Software Cost Estimation: Productivity; Estimation techniques; Algorithmic cost modeling, Project duration and staffing. **10 Hours**

Course Outcomes:

1. Describe the basics of Software engineering and code of conduct
2. Explain the various software production techniques (process models).
3. Apply rapid development tools in project development.
4. Analyze the difference between verification, validation, software evolution and software costing
5. Apply the management concepts for managing a software project.

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H		H	M		H			M			
2	H	H	H		H						H	M
3	H	H		H	H	H	H	M	M		M	
4	H	H	M		H		M	M	M		H	

TEXT BOOK:

1. Ian Sommerville: Software Engineering, 8th Edition, Person Education Ltd., 2007.

REFERENCE BOOKS:

1. Roger.S.Pressman: Software Engineering-A Practitioners approach, 7th Edition, McGraw-Hill, 2007.
2. Pfleeger: Software Engineering Theory and Practice, 2nd Edition, Pearson Education,2001.
3. Waman S Jawadekar: Software Engineering Principles and Practice, Tata McGraw Hill, 2004.

****

OPERATING SYSTEMS

Sub Code : 15CS505 **Credits : 04**
Hrs/Week : 4+0+0+S* **Total Hours : 52**

*** Self Study to be exercised under the supervision of course instructor and to be restricted to not more than 10% of the total teaching hours.**

Course Learning Objectives:

This Course will enable students to

1. Explain the concepts, principles and services of operating system.
2. Identify fundamental operating system abstractions such as Process, Threads, Files, Semaphores, IPC abstractions and demonstrate them.
3. Assess the benefits of concurrency and synchronization and apply them to write concurrent programs.
4. Analyze basic resource management technologies in job and process scheduling.
5. Use and compare different memory management techniques.
6. Study Linux Operating System in terms of process scheduling, Memory management, File system and I/O.

UNIT - I

Introduction to Operating Systems, System structures: Operating System structure; Operating System operations(functions), Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.

Process Management: Process concept; Process scheduling; Operations on processes; Inter-process communication.

Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; threading issues. Process - thread comparison. **10 Hours**

UNIT – II

Process Scheduling: Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; thread scheduling.

Process Synchronization

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

Deadlocks

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

10 Hours

UNIT – III

Memory Management: Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **10 Hours**

UNIT – IV

File System, Implementation of File System File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection.

Implementing File System: File system structure; File system implementation;

Directory implementation; Allocation methods; Free space management. **11 Hours**

UNIT - V

Secondary Storage Structures, Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management.

Self Study Component: The Linux & Windows Operating System : history; Design principles; Kernel modules; Process management; Scheduling; Memory management; File systems, Input and output; Inter-process communication - Comparison.

Tutorials: Students may be asked to go through the LINUX kernel code to understand the OS design. Students may be asked to implement a few algorithms related to operating system**.**

10 Hours

Note: Students must submit a report for the studies conducted & programs implemented at the end of semester.

Course Outcomes:

At the end of the course the student will be able to:

1. Explain basics of Operating Systems and OS Service.
2. Illustrate fundamentals of operating system concepts like process, threads, files, semaphores and IPC.
3. Explain the benefits of concurrency and synchronization and apply them to write concurrent programs.
4. Assess basic resource management technologies in job and process scheduling, Illustrate the deadlocks and management of deadlocks.
5. Explain the concepts of virtual and secondary memory management and various file system implementations.

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	**H**	**H**			**L**				**M**			
2	**H**	M			**H**				**H**		**L**	**H**
3	**H**				**H**							**H**
4	**H**	**H**			**M**				**L**		**L**	
5	**H**				**H**				**H**			

H : High M: Medium L : Low

TEXT BOOK:

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

REFERENCE BOOKS:

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

****

ADVANCED UNIX PROGRAMMING

Sub Code : 15CS511 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

At the end of the course student should be able to:

1. **List** the file APIs and **Write** and execute file handling programs.
2. **Explain** the concept of processes and its environment.
3. **Implement** programs to handle processes in Linux platform.
4. **Apply** inter process communication concept for data exchange between programs.
5. **Demonstrate** the concepts of signals and timers.

UNIT – I

The POSIX standards. File types. General File APIs, File handling programs. **7 Hours**

UNIT – II

Makefile – introduction, creation and execution of make file.
THE PROCESS: Introduction, Mechanism for creating process. The UNIX Kernel support for process.
THE ENVIRONMENT OF A UNIX PROCESS: Introduction, main function, Process Termination, Command line arguments, Environment List, Memory layout Of a C program, Memory allocation, Environment variables, functions. **8 Hours**

UNIT – III

Setjmp and longjmp functions, getrlimit, setrlimit
PROCESS CONTROL: Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, Wait3, wait4 functions, Race conditions, exec functions, Interpreter files, System Function.
SIGNALS: The UNIX Kernel Support for signals, Signal, **8 Hours**

UNIT - IV

Signal mask, Sigaction, The SIGCHLD Signal and waitpid functions,The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX .1b Timers.
DAEMON PROCESSES: Introduction, Daemon Characteristics, Coding Rules. **8 Hours**

UNIT – V

INTERPROCESS COMMUNICATIONS: Overview of IPC Methods, Pipes, popen, Pclose functions, FIFOs, Message Queues, Semaphores, Shared Memory.
SOCKETS: Introduction, functions, Client/Server Message Handling Example. **8 Hours**

Course Outcomes:

1. **Define** and discuss the POSIX standard and different types of files.
2. **Explain and apply** various APIs for file handling.
3. **Illustrate** the representation of a process and its environment and **apply** various process APIs for handing the processes

4. **Explain** the concept of signal and its handling methods. **Use** the signal handling APIs in programs.
5. **Describe and implement** the concepts of demon process and inter process communication.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1		M										H
2		H			M						L	H
3		H			M							H
4		H			M						L	H
5		H			M						L	H

H : High M: Medium L : Low

TEXT BOOKS:

1. Terrence Chan: UNIX System Programming Using C++, Prentice Hall India,1999.
2. W.Richard Stevens: Advanced Programming in the UNIX Environment, Addison – Wesley/PHI
3. Sumitaba Das, UNIX-Concepts and Applications, 4th Edition, Tata McGraw Hill, 2006. (Chapter 9).

REFERENCE BOOKS:

1. Maurice.J.Bach:The Design of the Unix Operating System, Pearson Education /Prentice Hall of India.
2. Uresh Vahalia: UNIX Internals, Pearson Education, ASIA, 2001.
3. R. Stones, N. Matthew, Beginning Linux Programming, Wrox publication.

****

SIGNALS AND SYSTEMS THEORY

Sub Code : 15CS512 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This course will enable the student to:

1. Explain the concept of signals.
2. Formulate the signal in the form of equations.
3. Represent the signal in Fourier form and apply this.
4. Perform Z transform on the signals.

UNIT - I

Introduction

Definitions of a signal and a system, classification of signals, basic operations on signals, elementary signals, systems viewed as interconnections of operations, properties of systems.

7 Hours

UNIT - II

Time-domain representations for LTI systems

Convolution, impulse response representation, properties of impulse response representation, differential and difference equation representations, block diagram representations. **8 Hours**

UNIT - III

Fourier representation for signals

Introduction, Fourier representations for four signal classes, orthogonality of complex sinusoidal signals, DTFS representations, continuous-tine-Fourier-series representations, DTFT and FT representations, properties of Fourier representations. **8 Hours**

UNIT - IV

Application of Fourier representations

Frequency response of LTI systems, solution of differential and difference equations using system function, Fourier transform representations for periodic signals, sampling of continuous time signals and signal reconstruction. **8 Hours**

UNIT - V

Z-Transforms

Introduction, Z-transform, properties of ROC, properties of Z-transforms, inversion of Z-transforms, transforms analysis of LTI systems, transfer function, stability and causality, unilateral Z-transforms and its application to solve difference equations **8 Hours**

Course Outcomes:

At the of the course the end student should be able to:

1. Classify different types of signals and systems.
2. Find the output of a LTI system.
3. Formulate the basic operations on signals.
4. Demonstrate system using differential/difference equation
5. Analyze signals & LTI systems in frequency & Z domain.

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H											
2	H											
3	H				M							
4	H				M							
5	H	M										

H : High M: Medium L : Low

TEXT BOOK:

1. **Simon Haykin and Barry Van Veen,** "Signals and Systems", John Wiley and Sons, 2001, Reprint 2002.
   Chapters : 1.1 to 1.8, 2.2 to 2.5, 3.1 to 3.6, 4.2 to 4.3, 4.7, 7.1 to 7.6, 7.8

REFERENCE BOOKS:

1. **Alan V. Oppenheim, Alan S. Willsky and S. Hamid Nawab,** "Signals and Systems", Pearson Education Asia, 2nd edition, 1997, Indian reprint 2002.
2. **Dr. D.ganesh Rao and Satish Tunga,** "Signals and Systems-A Simplified Approach", Sanguine Technical Publishers, 2003-04

****

DATA COMPRESSION

Sub Code : 15CS514 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. **Explain the** mathematical concepts behind compression.
2. **Develop** theoretical foundations of data compression, concepts.
3. **Develop** algorithms for lossy and lossless data compression, signal modeling.
4. **Apply** extensions of concepts learned to compression with applications to speech, image and video processing.

UNIT – I

Introduction, Lossless Compression -1

Compression techniques; Modeling and Coding.

Mathematical preliminaries for lossless compression: Overview; Basic concepts of Information Theory; Models; Coding; Algorithmic information theory; Minimum description length principle.

Huffman coding: Overview; The Huffman coding algorithm, Minimum variance Huffman codes; Application of Huffman coding for text compression. **7 Hours**

UNIT – II

Lossless Compression -2

Dictionary Techniques: Overview; Introduction; Static dictionary; Adaptive dictionary; Applications: **UNIX** compress, **GIF, PNG, V.42.**

Lossless image compression: Overview; Introduction; Basics; CALIC; JPEG-LS; Multiresolution approaches; Fascimile encoding: Run-length coding, T.4 and T.6. **8 Hours**

UNIT – III

Basics of Lossy Coding

Some mathematical concepts: Overveiw; Introduction; Distortion criteria; Models. Scalar quantization: Overview; Introduction; The quantization problem; Uniform quantizer; Adaptive Quantization.

Vector Quantization, Differential Encoding

Vector quantization: Overview; Introduction; Advantages of vector quantization over scalar quantization; The LBG algorithm.

Differential Encoding: Overview; Introduction; The basic algorithm; Prediction in DPCM; Adaptive DPCM. **8 Hours**

UNIT – IV

Some mathematical Concepts, Transform coding

Some mathematical concepts: Linear Systems; Sampling; Discrete Fourier transform; Z-transform.

Transform coding: Overview; Introduction; The transform; Transforms of

interest; Quantization and coding for transform coefficients.

Subband Coding, Audio Coding

Subband coding: Overview; introduction; Filters; The basic subband coding algorithm; Bit allocation. Audio coding: Overview; introduction; MPEG audio coding. **8 Hours**

UNIT - V

Wavelet- Based Compression

Overview; Introduction; Wavelets; Multiresolution and the scaling function; Image compression; JPEG 2000.

Video Compression

Overview; Introduction; Motion compensation; Video signal representation; H.261; Model-based coding; Asymetric applications; **MPEG-1** and **MPEG-2**;H.263; H.264; **MPEG-4** and advanced video coding **8 Hours**

Course Outcomes:

1. **Express** mathematical preliminaries for lossless Compression and Huffman coding.
2. **Apply** lossless Compression such as UNIX Compress, GIF, PNG, V.42 and **identify** lossless image Compression.
3. **Describe** the basics of lossy coding, vector quantization and differential Encoding.
4. **Express** mathematical concepts of linear systems, sampling, TFT, Z-transform and **describe** the transform coding, sub-band coding and audio coding.
5. **Explain** the Wavelet-based Compression and Video Compression.

Mapping of POs & COs:

POs COs	a	b	c	d	E	f	g	h	i	j	k	l
1	H		M		M							
2	H	M			H			L			M	H
3	M	M	M		H							L
4	L				M					H	H	M
5	L	H	L									M

H : High M: Medium L : Low

TEXT BOOK:

1. Khalid Sayood: Introduction to Data Compression, 3rd edition, Elsevier, 2006.
(Chapters1,2 (excluding 2.2.1 and 2.4.3), 3.1, 3.2, 3.2.1, 3.8.2, 5, 7.1 to 7.5, 7.6, 7.6.1, 7.6.2, 8.1 to 8.3, 8.6, 9.1 to 9.5, 10.1 to 10.4, 11.1 to 11.5, 12.6 to 12.9, 13.1 to 13.5, 14.1 to 14.4, 14.9, 15.1 to 15.4, 15.6, 15.9, 16.1 to 16.3, 18.1 to 18.12)

REFERENCE BOOK:

1. D. Salomon: Data Compression: The complete Reference, Springer, 1998.

OPERATIONS RESEARCH

Sub Code : 15CS515 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

UNIT – I

Introduction

Introduction to OR, nature and meaning, applications, modeling in OR, phases of OR study

Linear Programming

Introduction to Linear Programming through an example, graphical method ,formulation of LP model from practical problems, assumptions and properties of linear programming, simplex method **7 Hours**

UNIT – II

Revised simplex method, Big M method, 2 phase method, Duality theory, Primal and dual relationship, Dual simplex method **8 Hours**

UNIT - III

Transportation Problems:

Special types of main programming, transportation problems, methods to find initial feasible solution and modification to obtain optimal solution) Degeneracy in transportation problems, unbalanced transportation problems **8 Hours**

UNIT – IV

Assignment problem

Mathematical formulation of an assignment problem, unbalanced assignment problem ,TSP, Hungarian method **8 Hours**

UNIT – V

CPM ,PERT

Representation of a project by a network, activities and events, starting times, finishing times, floats, slacks, CPM, Idea of crashing probabilistic times and PERT analysis **8 Hours**

Course Outcomes:

1. Discuss the basics of OR, modelling and application of OR.
2. Define linear programming model. Formulate linear programming model and apply linear programming model using different techniques.
3. Formulate the problem and solve the problem by using different techniques.
4. Describe mathematical formulation of an Assignment Problem and solve various scenarios by using different methods.
5. Identify the project management techniques.

Mapping of POs & COs :

POs / COs	a	b	c	D	e	f	g	h	i	j	k	l
1	L											
2	H				H							
3	H				H							
4	H		M		H							
5					M					H		

TEXT BOOKS:

1. Operations Research, S D Sharma, 15th edition
2. Operations Research – An introduction, Hamdy A Taha, PHI, 7th edition

REFERENCE BOOKS:

1. Operation research, Kantiswaroop. Manmohan and Gupta
2. Introduction to operation research, a computer oriented algorithmic approach, Gillelt B G, McGraw Hill, 1976

****

PROGRAM VERIFICATION

Sub Code : 15CS516 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives (CLO)

After studying this subject, the student should be able to:

1. ***Apply*** the mathematical and logical concepts for programming.
2. ***Explain*** various Programming paradigms.
3. ***Indentify*** the specifications of a sequential program.
4. ***Write s***imp***l e*** program using Dafny.
5. ***Perform*** program verification using Dafny.

UNIT I

BACKGROUND AND INTRODUCTION:

Sequential, concurrent, and reactive systems, Programming languages and paradigms, Type systems of programming languages, Assigning meaning to programs, operational semantics denotational semantics, Partial and total correctness, Hoare triples, Logic for Program Design **:** Propositional Calculus, Predicate Calculus.

6 Hours

UNIT II

MATHEMATICAL AND LOGICAL FOUNDATIONS:

Mathematics for Specification**:** Sets, Relations, Functions and Sequences. Pre conditions, Post conditions Loop invariants. **7 Hours**

UNIT III

SPECIFICATION OF PROGRAMS:

Variant functions, the state model of programs, Partial and total correctness, Weakest precondition, Guarded commands, Why functional programming matters, Algebraic data types, Higher order functions. **8 Hours**

UNIT IV

PROGRAM VERIFICATION USING DAFNY PART-I

Methods and functions, pre and post conditions, Assertions, loop invariants, termination, quantifiers, framing, Binary search—an example. 9 **Hours**

UNIT V

PROGRAM VERIFICATION USING DAFNY PART-II:

Predicates, sets, sequences, collections, Lemmas, modules: Declaring a new module, Import and export new module, opening modules. **9 Hours**

Course Outcomes

At the end of the course the student will be able to:

1. Identify various paradigms related to programming.
2. Build logical and mathematical specifications for program
3. Testing for specifications of the program.
4. Design and write simple Dafny programs and learn basic syntax.
5. Apply advanced Dafny tool for program verification.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	F	g	h	i	j	k	l
1	L	L									H	H
2	H	L			M							L
3	L										H	H
4		M			L						H	H
5	L	L			L						H	M

H : High M: Medium L : Low

TEXT BOOKS, REFERENCES, AND ONLINE RESOURCES:

1. Geoff Dromey, Program Derivation. International Computer Science Series. Addison-Wesley. 1989.
2. Michael Huth and Mark Ryan. Logic in Computer Science - Modeling and Reasoning about Systems. Cambridge University Press. 2004.
3. Dafny: a language and program verifier for functional correctness. http://research.microsoft.com/en-us/projects/dafny/, Microsoft Research.

INTRODUCTION TO INTERNET OF THINGS

Sub Code : 15CS517 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Objectives: This course will enable students to

1. Explain the IOT concepts and applications
2. Describe Security and Privacy issues in IOT
3. Explain IOT Data management and Standards
4. Discuss IOT requirements and Smart Objects
5. Explain the cloud and Integration services to IOT

UNIT – I

Introduction, **Putting the Internet of Things Forward to the Next Level,** The Internet of Things Today, The Internet of Things Tomorrow, Potential Success Factors, **Internet of Things Strategic Research and Innovation Agenda** - Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Smart-X Applications. **8 hours**

UNIT – II

Internet of Things and Related Future Internet Technologies, Networks and Communication, Processes, Data Management, Security, Privacy & Trust, Device Level Energy Issues, IoT Protocols Convergence, I**nternet of Things Global Standardization - State of Play –** Introduction, IoT Vision, IoT Standardization Landscape, **Dynamic Context-Aware Scalable and Trust-based IOT** **8 hours**

UNIT- III

Security, Privacy Framework – Introduction, Main Concepts and Motivation of the Framework,, A Policy-based Framework for Security and Privacy in Internet of Things

8 hours

UNIT- IV

Scalable Integration Framework for Heterogeneous Smart Objects, Applications and Services – Introduction, IPv6 Potential, IoT6, IPv6 for IOT, Adapting IPv6 to IoT Requirements, IoT6 Architecture. **8 hours**

UNIT – V

DigCo Integration with the Cloud and EPICS, Enabling Heterogeneous Integration, IoT6 Smart Office Use-case, Scalability Perspective, Conclusions. **7 Hours**

Course Outcomes: After Studying this course, students will be able to

1. Demonstrate understanding of IOT Concept and Applications
2. Analyze Security and Privacy Framework issues in IoT
3. Apply IOT Data management and Standards
4. Analyze IOT requirements and Smart Objects
5. Apply the cloud and Integration services to IOT

Text Books:

Internet of Things From Research Innovation to Market Development, Dr, Ovidiu Vermesan, SINTEF Norway, Dr. Peter Friess, EU Belgium, River Publishers, Aalborg

Reference Books:

The Definitive Guide to the Internet of Things for Business, 2nd Edition, By Syed Zaeem Hosain, CTO, Aeris

Mapping COs & POs : Course Learning Objectives:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1		H		L								
2	H				H					H	M	H
3			H		M	H						
4		H			H					H	H	

****

COMPUTER GRAPHICS AND MULTIMEDIA

Sub Code : 15CS601 **Credits : 04**
Hrs/Week : 4+0+2+0 **Total Hours : 52**

Course Learning Objectives:

This Course will enable students to

1. **Explain** the concepts of application of graphics, **Illustrate** the theory behind the graphics hardware, their **design** and to **differentiate** raster and vector display devices. **Get** the idea of different raster graphics algorithm used and **tell how** they are used by graphics packages.
2. **Illustrate** different geometrical transformations in 2D and 3D and to **make use of** transformations to solve different mathematical problems. **Outline** different color models for raster graphics.
3. **Illustrate** different Fractal Geometry methods, illumination models, shading models for polygons and **find** out the differences between them.
4. **Outline** the theory behind different visible surface algorithms and to **identify** the advantages and disadvantages of a particular technique
5. **Get** the idea of different openGL functions and to **make use of** openGL libraries for programming different graphical models.

UNIT – I

INTRODUCTION:

Introduction to graphics, Raster and random scan displays, video controller, Applications of Computer Graphics. **2 Hours**

RASTER GRAPHICS ALGORITHM

Scan converting lines & circles: Midpoint algorithm, Filling rectangles, Filling Polygons, Clipping lines: Cohen Sutherland, Liang Barsky algorithms, Clipping polygons: Sutherland-Hodgeman algorithm, Antialiasing **8 Hours**

UNIT - II

GEOMETRICAL TRANSFORMATIONS (2D):

2D Transformations, Homogeneous coordinates and Matrix representation of 2D Transformations, composition of 2D Transformations. The window to view port transformation. **2 Hours**

GEOMETRICAL TRANSFORMATIONS (3D):

Matrix representation of 3D Transformations, Transformations as change in coordinate system. **3 Hours**

VIEWING IN 3D:

3D Viewing Process, Specification of an Arbitrary 3D View, Types of projections, Color Models for Raster Graphics. **5 Hours**

UNIT - III

CURVES, FRACTALS AND SHADING:

Polygon surfaces, curved lines and surfaces, Quadratic surfaces, Spline Representation, Bezier & B-Spline Curves & Surfaces, Fractal Geometry methods, Illumination models, Shading models for polygons , surface details and shadows**.** **10 Hours**

UNIT – IV

VISIBLE SURFACE DETERMINATION:

Functions of two variables, Techniques for efficient Visible surface Algorithms, Algorithms for visible line determination. The Z-buffer Algorithm, List priority Algorithms, scan-line Algorithms, Area- subdivision Algorithms, Algorithms for curved surfaces, Visible-surface Ray tracing. **10 Hours**

UNIT - V

OPENGL:

General OpenGL Introduction, Input & Interaction, Viewing, Lighting, Shading, Hidden Surface removal, Discrete techniques, Programmable Shaders, Modeling & hierarchy.[Only OpenGL functions] **12 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. **Identify** basic graphics, graphic devices and **illustrate** raster graphics algorithms.
2. **Explain** and **apply** various geometrical transformations including 2D and 3D and **simplify** different transformations.
3. **Design** the curves, fractal and **apply** shading for polygons and also the to **illustrate** different illumination techniques and their properties.
4. **Identify** and **apply** the techniques for various efficient visible surface determination algorithms.

5. **Apply** Open GL for graphics programming to **design** various graphical models and also to **construct** the real time animations

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1	M		M								L	
2		L	H		H						M	
3	M	L									L	M
4		M	M								L	H
5		H			H							H

H : High M: Medium L : Low

TEXT BOOKS:

1. Computer Graphics - Addision-welesly 1997 by - James D. Foley, Andries Van Dam, Steven K feiner, John F. Huges .
2. Computer Graphics – A top down approach with Open GL by Edward, Angel, Adison Wesley 2000.

REFERENCE BOOK:

1. OpenGL Programming Guide, Release 1 by Jackie Neider, Tom Davis, Mason Woo , Addison-Wesley Publishing Company.

****

COMPUTER GRAPHICS & MULTIMEDIA LABORATORY

Sub Code : 15CS601 **Credits : 01**
Hrs/Week : 2

Course Learning Objectives:

This Course will enable students to

1. **Demonstrate** the implementation of different raster algorithms, clipping algorithms,2D transformations and 3D models using programming languages like C/C++.
2. **Make use of** different Open GL API's and to **demonstrate** their usage through the graphical models.
3. **Propose** and **develop** a graphical mini project using any language.

A. Student has to write and execute programs in C/C++ using OPENGL on Windows/Linux platform to implement a few graphics applications like:

1. Transformations in both 2D and 3D
2. Clipping
3. 3D viewing
4. Hidden line removal
5. Fractal generation

B. Student may also be asked to implement one or two graphics algorithms like Line drawing or Circle drawing or Filling by using only graphic primitives

C. Graphics Mini project implementation.

Course Outcomes:

At the end of the course the student will be able to:

1. **Experiment** different graphics algorithm design technique and **analyze** their efficiency.
2. **Design** and **experiment** graphics algorithms using only graphics primitives.
3. **Design** and **extend** a graphics mini project using OpenGL

Mapping of POs & COs:

POs COs	**a**	**b**	**c**	**d**	**E**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
1		**H**			**M**							**M**
2		**H**			**H**							**M**
3		**H**			**H**							**M**

H : High M: Medium L : Low

****

COMPUTER NETWORKS

Sub Code : 15CS602 **Credits : 04**
Hrs/Week : 4+0+2+0 **Total Hours : 52**

Course Learning Objectives
This Course will enable students to

1. Outline the principles of computer networks and its application
2. Illustrate the concept of types of network
3. Identify the issues in network layer and solution for it
4. Analyze the process of congestion control algorithms
5. Illustrate IP Packets and fragmentation process

UNIT – I

INTRODUCTION TO COMPUTER NETWORKS:
Uses of computer networks : Business Applications, Home Application, Mobile Users, Social Issues; Network hardware: Local Area Networks, Metropolitan Area Networks, Wide Area Networks, Wireless Networks, Home Networks, Internet works; LAN Protocols : LAN Structure , The Medium Access Control Sub layer, The Logical Link Control Sublayer; Ethernet and IEEE 802.3 LAN Standard : Ethernet Protocol, Frame structure, Physical Layers, Fast Ethernet ,Gigabit Ethernet, Gigabit Ethernet; Token Ring and IEEE 802.5 LAN Standard: Token-Ring Protocol, Frame structure; FDDI; **10 Hours**

UNIT – II

NETWORK LAYER (PART-I) :
Network layer design issues: Store and Forward packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection-Oriented Service, Comparison of Virtual Circuit and Datagram Subnets; Routing algorithms: The Optimality Principal , Shortest Path Routing, Flooding. Distance Vector Routing, Link state Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing , Routing for Mobile Hosts, Routing in Ad hoc Networks. **10 Hours**

UNIT – III

NETWORK LAYER (PART – II) :
Congestion Control Algorithms: General Principles of Congestion Control, Congestion Prevention Policies, Congestion Control in Virtual-Circuit Subnets, Congestion Control in Datagram Subnets, Load Shedding, Jitter Control; Quality Of Service: Requirements, Techniques for Achieving Good Quality of Service, Integrated Services, Differentiated Services; **11 Hours**

UNIT – IV

NETWORK LAYER (PART – III)
Internetworking: How networks differ, How Networks Can Be Connected, Concatenated Virtual Circuits, Connectionless Internetworking, Tunneling, Internetwork Routing,

Fragmentation; The Network Layer in the Internet : The IP Protocol, IP Addresses, Internet Control Protocols, The Interior Gateway Routing Protocol : OSPF, The Exterior Gateway Routing Protocol : BGP, Internet Multicasting, Mobile IP, Ipv6; **11 Hours**

UNIT – V

THE TRANSPORT LAYER

The Transport Service: Services Provided to the Upper Layers, Transport Service Primitives, Elements of Transport Protocols: Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing, Crash Recovery; The Internet Transport Protocols(UDP):Introduction to UDP, The Internet Transport Protocols(TCP): Introduction to TCP, The TCP Service Model, The TCP Protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release, Modeling TCP Connection Management, TCP Transmission Policy, TCP Congestion Control. The Application Layer: DNS- Domain Name System, The WWW, Static Web documents, Dynamic Web documents. **11 Hours**

Course Outcomes

At the end of the course the student will be able to:

1. Express the basic concept of computer network .
2. Design the network layer and the related issues .
3. Explain the congestion control, and prevention methods .
4. Explain different type of networks and protocols..
5. Describe the transport layer protocols .

Mapping of POs & COs :

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H			H				L	L		H	
2		H		H	H	H		L		M	M	H
3	H		H		H	H		L		M	H	H
4		H			H	L		L		M		H
5	H	H			H	H		L	L	M		H

H : High M: Medium L : Low

TEXT BOOKS

1. Andrew S. Tanenbaum , Computer Networks, Fourth edition, PHI / Pearson Publication, 2002 .
2. Alberto Leon – Garcia and Indra Widjaja , Communication Networks – Fundamental Concepts and Key architectures, Tata McGraw-Hill 2nd edition . (Chapter 6)

REFERENCE BOOKS

1. Behrouz A. Forouzan , Data Communications and Networking, Tata McGraw- Hill 3rd Edition.
2. William Stalling , Data and Computer Communication, Fifth Edition, Prentice Hall India.
3. James F. Kurose and Keith W. Ross , Computer Networking- A Top-Down Approach Featuring the Internet, 2nd Edition, Pearson .
4. Larry L. Peterson and Bruce S. Davie, Computer Networks, 3rd Edition, Elsevier –Morgan Kaufmann Publishers.

****

COMPUTER NETWORKS LABORATORY

Sub Code : 15CS602 **Credits : 1**
Hrs/Week : 2

LIST OF PROGRAMS

Write a program for simple RSA algorithm to encrypt and decrypt the data.

Write a program for error detecting code using CRC-CCITT (16-bits).

Write a program for Hamming Code generation for error detection and correction.

Write a program for frame sorting technique used in buffers.

Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Write a program for distance vector algorithm to find suitable path for transmission.

Write a program for congestion control using Leaky bucket algorithm.

1. Simulate a three nodes point-to-point network with duplex links between them. Set the queue size vary the bandwidth and find the number of packets dropped.
2. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3 and UDP n1-n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets by TCP/UDP.
3. Simulate the different types of Internet traffic such as FTP a TELNET over a network and analyze the throughput.
4. Simulate the transmission of ping messaged over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
5. Simulate an Ethernet LAN using N-nodes(6-10), change error rate and data rate and

compare the throughput.

6. Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and determine collision across different nodes.

7. Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and plot congestion window for different source/destination.

****

JAVA AND INTERNET TECHNOLOGIES

Sub Code: 15CS603 **Credits : 04**
Hrs/Week : 4+0+2+0 **Total Hours : 52**

Course Learning Objectives:

At the end of the course student should be able to:

1. **Design** an object oriented logic for a given problem scenario.
2. **Describe** various programming constructs that are used in core Java.
3. **Develop** Java User Interface programs by using the AWT and Swing APIs
4. **Analyze** real time problems and develop multi-threaded java programs to solve.
5. **Develop** networking Java applications.
6. **Apply** the database concept for a Java database applications
7. **Design** server side web applications using Java Servlets.
8. **Develop** server side web applications using Java Server Pages.

UNIT - I

Introducing Classes – Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, this keyword, Method overloading, Using objects as parameters, Argument passing, Returning objects, Access control, static, final, Using command line arguments, variable length arguments.

Inheritance – Inheritance Basics, Using super, creates a Multilevel Hierarchy, When constructors are called? Method Overriding, Using abstract classes, Using final with Inheritance.

Packages and Interfaces – Packages, Access protection, Importing Packages, Interfaces.

Exception Handling – Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, multiple catch Clauses, Nested try statements, throw, throws, finally. **12 Hours**

UNIT - II

Multithreaded Programming – The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Inter-thread Communication.

Event Handling - Two Event Handling Mechanisms, The Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Event Model.

Working with Windows, Graphics and Text - Overview, AWT Classes, Window Fundamentals, Working with Frame Windows, Creating a Frame Window, Creating a Windowed Program, Displaying Information within a Window, Working with Graphics, Working with color. **10 Hours**

UNIT - III

Swings – Swing key features, MVC connection, components and containers, Event handling, Painting, Simple examples; Exploring Swings, Swings UI components.

Input/Output – I/O Basics, Reading Console Input, Writing Console Output, The Print Writer Class.

File Handling - Serial Access Files, File Methods, Redirection, Command Line Parameters, Random Access Files. **10 Hours**

UNIT - IV

Java Database Connectivity (JDBC) - The Vendor Variation Problem, SQL and Versions of JDBC, Creating an ODBC Data Source, Simple Database Access, Modifying the Database Contents, Transactions, Meta Data, Scrollable Result Sets in JDBC 2.0, Modifying Databases via Java Methods.

Network Programming with Java - Basic Concepts, Protocols and Terminology, Clients, Servers and Peers, Ports and Sockets, The Internet and IP Addresses, Internet Services, URLs and DNS, TCP, UDP. The Inet Address Class, Using Sockets (TCP and UDP). **10 Hours**

UNIT - V

Java Servlets – Benefits, A simple Java Servlet, Anatomy of a Java Servlet, Reading data from a client, Reading HTTP Request Headers, Sending data to a client and writing the HTTP Response Header, Working with Cookies, Tracking Sessions.

Java Server Pages (JSP) – Installation, JSP Tags, Request String, User Sessions, Cookies, Session objects. **10 Hours**

Course Outcomes:

1. **Identify and apply** the basic concepts of Java programming, inheritance, packages & interfaces and exception handling.
2. **Explain** the multithreaded programming, event handling in Java.
3. **Design and Develop** user interfaces and programs using Java swings, input/output and file handling concepts.

4. **Apply** JDBC/ODBC connectivity and network programming in Java.
5. **Deign** a web page based on servlets and JSP.

Mapping of POs & COs:

P Os COs	a	b	c	d	e	f	g	h	i	j	k	l
1		H	H		H					L	L	H
2		H	M		H						M	H
3		H	H		H					L	M	H
4		H	H		H					L	M	H
5		H	H		H					L	M	H

H : High M: Medium L : Low

TEXT BOOKS:

1. The Complete Reference Java by Herbert Schildt, Seventh Edition, 2007, Tata McGraw-Hill.
2. An Introduction to Network Programming with Java by Jan Graba, 2007, Springer Publications.
3. The Complete Reference J2EE by Jim Keogh, Tata McGraw-Hill. 2002

REFERENCE BOOK:

1. Java – How to Program? by H. M. Deitel, 2004, Prentice Hall.

****

JAVA AND INTERNET TECHNOLOGIES LABORATORY

Sub Code : 15CS603 **Credits: 01**

Hrs/Week : 2

Course Learning Objectives:

At the end of the course student should be able to:

1. **Design** programs using object oriented concepts.
2. **Design and implement** User interfaces using swings.
3. **Design and implement** Database and network programs using Java.
4. **Develop** servlet and JSP programs
5. Java programs that includes each of the topic discusses in the theory.
6. Programs on Simple programs and programs to design simple user interface using HTML and users Servelets, JSP

INSTRUCTIONS:

1. In exam two programs may be asked randomly in any combination.
2. Any number of programs can be practiced in lab under each section.
3. Programs should incorporate as many features as possible.
4. Program need may be based on various features of the technology being used.
5. It is mandatory to score minimum marks both in Theory and Lab.

Course Outcomes:

1. Implement Java program that contains inheritance, event handling, packages & interfaces concepts.
2. Implement Java program that contains AWT and file handling, JDBC and networking concepts.
3. Design a simple user interface using HTML and implement programs on Java Script.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1		H			H					L	M	H
2		H			H					L	M	H
3		H			H					L	M	H

H : High M: Medium L : Low

****

SOFTWARE TESTING AND AUTOMATION

Sub Code : 15CS604 **Credits : 03**

Hrs/Week : 3+1+0+S* **Total Hours : 39**

*** Self Study to be exercised under the supervision of course instructor and to be restricted to not more than 10% of the total teaching hours.**

Course Learning Objectives:

This Course will enable students to

1. Explain the concept of testing and the testing life cycle.
2. Use the testing frameworks, process and test management to generate the test plans.
3. Generate the test plans for a business.
4. Illustrate the use of automation in testing.
5. Perform defect management and data management.

UNIT - I

Chapter-1: Introduction to Testing – why and what,
Why is testing necessary? What is testing? Role of Tester, Testing and Quality, Overview of STLC **3 Hours**

Chapter-2: Software Testing Life Cycle - V model**,**
SDLC vs STLC, different stages in STLC, document templates generated in different phases of STLC, different levels of testing, different types of testing **3 Hours**

UNIT - II

Chapter-3: Static Testing**, -**
Static techniques, reviews, walkthroughs **2 Hours**

Chapter-4: Basics of test design techniques**, -**
Various test categories, test design techniques for different categories of tests. Designing test cases using MS-Excel. **5 Hours**

UNIT - III

Chapter-5: Test management**, -**
Documenting test plan and test case, effort estimation, configuration management, project progress management. Use of Testopia for test case documentation and test management. **4 Hours**

Chapter-6: Defect management**, -**
Test Execution, logging defects, defect lifecycle, fixing / closing defects. Use of Bugzilla for logging and tracing defects. **4 Hours**

UNIT - IV

Chapter-7: Test Data Management**, -**
Test Data Management –Overview, Why Test Data Management, Test Data Types, Need for Test Data Setup, Test Data Setup Stages, Test data management Challenges. Creating sample test data using MS-Excel **3 Hours**

Chapter-8: Basics of Automation testing**, -**
Introduction to automation testing, why automation, what to automate, tools available for automation testing. **4 Hours**

UNIT - V

Chapter-9: Basics of Automation testing using Selenium**,**
Introduction to Selenium, using Selenium IDE for automation testing, using Selenium Web driver for automation testing, understanding TestNG framework with Selenium Web driver for automation testing **11 Hours**

REFERENCE BOOKS:

1. Rex Black (2001), Managing the Testing Process (2nd edition), John Wiley & Sons
2. Foundations of software testing – by Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black
3. Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality - By: Elfriede Dustin
4. Selenium.org - http://docs.seleniumhq.org/docs/
    - Need following .XPI and .jar files to be downloaded from http://www.seleniumhq.org/download/
        - Selenium-ide.xpi (latest current version)

- Selenium-java.jar (latest current version)
- Firebug.xpi (latest current version)
- Firepath.xpi (latest current version)
- IEDriverServer.exe (latest current version)
- Jxl.jar (latest current version

Course Outcomes:

At the end of this elective, student will be able to:

1. Explain the complete software testing life cycle and Demonstrate understanding of various terms and technologies used in testing domain
2. Demonstrate the usage of testing framework, process and test management, generating test plan and designing test cases and test management process
3. Identify and write the test plan for a given a business scenario, design test cases, document test cases using an open source test management tool
4. Demonstrate understanding of defect management life cycle and ability to use an open source tool for defect management
5. Demonstrate understanding of test data management and automation testing

Mapping of POs & COs:

POs **COs**	**a**	**b**	**c**	**d**	**E**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
1	H	H	L		M					L	H	M
2	H	M	M		L					L	H	M
3	M	H	H									
4	M	H	H		L							
5	L	M	H		L					L	H	M

H: High M: Medium L : Low

****

ENTRY EDGE: IMMERSIVE GROUP WORKSHOP (IGW)

Sub Code: 15CS605 **Duration: 5 Days**

Timings : 9.00 AM to 12.30 PM, 1.15 PM to 4.45 PM

Module 1: Minds-on and hands-on simulation project

1. Understanding Task environment – Goals, responsibilities, Task focus
2. Working in Teams towards common goals
3. Organizational performance expectations–technical and behavioural competencies.

5 Hours

Module 2: Re- enforcement of critical individual skills and behaviours

1. Application of individual effectiveness skills in team and organizational context – improving self awareness, goal setting, time management, communication and presentation skills.

7 Hours

Module 3: Etiquettes and Ethics

1. Professional etiquettes at workplace – dressing, telephone, e-mail, meeting and general behaviour
2. Basic honesty & respect for law / rules
3. Conflict of interest
4. Use of organizational resources
5. Misrepresentation and misappropriation
6. Intellectual property
7. Whistle blowing **7 Hours**

Module 4: Interpersonal Behaviour & relationship skills

1. Establishing trust based relationships in team & organizational environment
2. Trust equation – credibility, responsiveness, integrity, self-interest **3.5 Hours**

Module 5: Dealing with Conflicts

Orientation towards conflicts in team and organizational environment

1. Understanding sources of conflicts
2. Conflict resolution styles and techniques **3.5 Hours**

<u>Pedagogical tools & techniques used in the workshop</u>

Organizational templates for simulating a organizational context- structures, units, roles and activities

Metaphoric scenarios for simulating real –life tasks and dynamics in a team/project context

LEGO ^{TM} building blocks for simulating last-mile technical activity in teams

Case studies, Role play scenarios group learning activities, observation and feedback.

Note: Evaluation is done and a grade of P (pass) or NP (not pass) is awarded

****

PATTERN RECOGNITION

Sub Code : 15CS611 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Credits : 39**

Course Learning Objectives:

This Course will enable students to

1. Explain the concepts Machine Perception, Pattern Recognition, Design cycle, learning and Bayesian Decision Theory2
2. Explain the concepts Machine Perception, Pattern Recognition, Design cycle, learning and Bayesian Decision Theory
3. Perform likelihood estimation, parameter estimation and complex analysis, Demonstrate nearest neighbour rule, metrics and nearest-neighbour classification and fuzzy classification
4. Explain the linear discriminant functions, Perceptron criterion function and squared-error procedures
5. Apply the principles of Learning, clustering, component analysis and multidimensional scaling

UNIT – I

Introduction: Machine Perception, Pattern Recognition systems, Design cycle, learning and adaptation (1.1, 1.3, 1.4, 1.5 of Ref.1)

Bayesian Decision Theory: Introduction, Bayesian Decision theory – continuous features, classifiers, discriminant functions, and decision surfaces, normal density and discriminant functions, Bayes decision theory – discrete features (2.1, 2.2, 2.4, 2.5, 2.6, 2.9 of Ref. 1)

8 Hours

UNIT - II

Maximum likelihood and Bayesian parameter estimation: Introduction, maximum likelihood estimation, Bayesian Estimation, Bayesian parameter estimation, problem of dimensionality, sufficient and exponential family, complex analysis & discriminants, (3.1 to 3.8 of Ref.1)

8 Hours

UNIT - III

Nonparametric Techniques: Introduction, Density Estimation, Parzen Windows, kn-nearest neighbour estimation, nearest neighbour rule, metrics and nearest-neighbour classification, fuzzy classification, reduced coulomb energy, approximations by series expansions (4.1 – 4.9 of Ref.1) **8 Hours**

UNIT - IV

Linear discriminant functions: Introduction, linear discriminant functions, generalized linear discriminant functions, minimizing the Perceptron criterion function, relaxation procedures, non separable behaviours, minimum squared-error procedures, Ho-Kashyap procedures (5.1 to 5.9 of Ref.1 **8 Hours**

UNIT - V

Unsupervised learning and clustering: Mixture densities and identifiability, maximum-likelihood estimates, application to normal mixtures, unsupervised Bayesian learning, data decryption and clustering, criterion functions and clustering, hierarchical clustering, on-line clustering. Component analysis, low-dimensional representations and multidimensional scaling (10.1 to 10.14 except 10.8, 10.12 of Ref. 1) **7 Hours**

Syntactic pattern Recognition: Overview, qualifying structure in pattern description and recognition, grammar based approach, elements of formal grammar (Chap. 3 of Ref. 2)

Course Outcomes:

1. Recall the basics of pattern recognition systems and Bayesian Decision Theory.
2. Determine the maximum likelihood and Bayesian parameter estimation.
3. Express the nonparametric techniques such as density estimation and nearest neighbour estimation.
4. Examine linear discriminant functions, minimizing the perception criterion function and minimum squared-error procedures
5. Describe the various unsupervised learning and clustering methods.

Mapping of POs & COs:

POs COs	a	b	c	d	E	f	g	h	i	j	k	l
1	H				M				L			
2	H	M	L		H						M	
3	H	M			H				L		M	
4	H		L		H				L		M	L
5	H		L		H			M	M		M	L

H : High M: Medium L : Low

TEXT BOOKS :

1. Richard O. Duda, Peter E. Hart and David G Stork, Pattern Classification, John Wiley & Sons, Inc.2nd Ed. 2001.
2. Robert Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches, John Wiley & Sons, Inc.1992.

****

SYSTEM SIMULATION & MODELING

Sub Code : 15CS612 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. **Brief** the appropriateness of the Simulation, its application, types of simulation model steps in simulation study and general principles in simulation.
2. **Illustrate** the use of Random-Number Generation and Random-Variate Generation techniques.
3. **Highlight** the issues connected with input modeling.
4. **Indentify** various Verification And Validation Of Simulation Models
5. **Discuss** some of issues associated with Simulation Of Computer Systems

UNIT - I

Introduction To Simulation:

When Simulation is the Appropriate Tool; When Simulation Is Not Appropriate; Advantages and Disadvantages of Simulation; Areas of Application; Systems and System Environment; Components of a System; Discrete and Continuous Systems; Model of a System; Types of Models; Discrete-Event System Simulation; Steps in a Simulation Study.

2. General Principles:

Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling. **8 Hours**

UNIT - II

3. Random-Number Generation:

Properties of Random Numbers; Generation of Pseudo-Random Numbers; Techniques for Generating Random Numbers; Tests for Random Numbers.

4. Random-Variate Generation:

Inverse Transform technique: Exponential Distribution, Uniform Distribution, Discrete Distributions; Acceptance-Rejection Technique: Poisson Distribution. **6 Hours**

UNIT - III

5. Input Modeling:

Data Collection; Identifying the distribution with Data; Parameter Estimation; Goodness of Fit Tests; Selecting Input Models without Data; Multivariate and Time-Series Input Models.

8 Hours

UNIT - IV

6. Verification And Validation Of Simulation Models:

Model Building, Verification and Validation; Verification of Simulation Models; Calibration and Validation of Models **8 Hours**

UNIT - V

7. Simulation Of Computer Systems:

Introduction; Simulation Tools; Model Input; High-Level Computer-System Simulation; CPU Simulation; Memory Simulation. **8 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. **Find and summarize** the suitability of the Simulation, its application, types of simulation model steps in simulation study and general principles in simulation.
2. **Apply** the Random-Number Generation and Random-Variate Generation techniques in practice.
3. **Tell** the issues connected with input modeling
4. **Make use of** various Verification And Validation Of Simulation Models
5. **Outline** some of issues associated with Simulation Of Computer Systems

Mapping of POs & COs:

POs Cos	a	b	c	d	e	f	g	h	i	j	k	l
1	M							M				
2		M	H		H						M	
3		L								M		M
4	M				M					H		
5	L	M							H		M	

H : High M: Medium L : Low

TEXT BOOK:

1. Jerry Banks, John S. Carson, Barry L. Nelson, David M. Nicol, "Discrete-Event System Simulation",Third Edition, Prentice-Hall India

REFERENCE BOOKS:

1. Averill M. Law, W. David Kelton, "Simulation Modeling and Analysis" ,Third Edition, McGrawHill.
2. Geoffrey Gordon, "System Simulation", Second Edition, Prentice-Hall India.

****

DISTRIBUTED SYSTEMS

Sub Code : 15CS613 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. **Identify** the issues involved in designing distributed systems.
2. **Describe** various synchronization methods of distributed methods.
3. **Analyze** process migration approach and distributed deadlock management.
4. **Describe** features distributed shared memory and file system.
5. **List** and describe load balancing mechanisms in distributed systems.

UNIT – I

Introduction to Distributed Systems:

Fundamentals:- What is Distributed Computing Systems?, Distributed Computing System Models, What is DOS?, Issues in designing a DOS.

Remote Procedure Calls: The RPC model, Transparency of RPC, Implementing RPC mechanism, Stub generation, RPC messages, Marshaling Arguments and results. Server management , Parameter passing semantics, call semantics, communication protocols RPC's. Complicated RPCs , Client –server binding, Exception handling, Security. **7 Hours**

UNIT – II

Synchronization in distributed Systems: Clock synchronization – logical clocks – physical clocks – clock synchronization algorithms, Mutual exclusion – A centralized algorithm – A distributed algorithm – a token ring algorithm, Comparison of the three algorithms, Election algorithms – the Bully algorithm – ring algorithm, Dead locks in distributed systems – distributed deadlock avoidance algorithms – distributed deadlock prevention algorithms,

distributed deadlock detection algorithms: Centralized approach, Hierarchical approach and Fully distributed approach. **8 Hours**

UNIT - III

Process Migration:
Desirable Features of a Good Migration Mechanism, Process Migration Mechanisms,
Threads: Introduction, Motivation for using Threads, Models for Organizing Threads, Issues in Designing Threads Package, Implementing Thread Package **8 Hours**

UNIT – IV

Distributed Shared Memory: General structure, Design and implementation issues of DSM, Granularity, Structure of shared memory space, Consistency Models, Replacement Strategy, Thrashing
Distributed File Systems:
Desirable features of a good distributed file system, file models, file accessing models, file sharing semantics, file Replication. **8 Hours**

UNIT – V

Resource Management: Desirable features, task management approach, load balancing approach, load sharing approach.
Naming: Introduction, Desirable Features of Good Naming System, System-Oriented Names, Object-Location Mechanism, Human Oriented Names, Name Caches **8 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. **Identify** the issues involved in designing distributed systems.
2. **Demonstrate** various synchronization methods of distributed methods.
3. **Compare various** process migration approaches and distributed deadlock management approaches.
4. **Apply** features distributed shared memory and file system.
5. **Evaluate** load balancing mechanisms in distributed systems.

Mapping of POs & COs:

P Os COs	a	b	c	d	e	f	g	h	i	j	k	l
1			H							L		
2			H									
3		M	H									
4		M	H		L							
5					H			L			M	

H : High M: Medium L : Low

TEXT BOOK:

1. Distributed Operating Systems, Concepts & Design, Pradeep K Sinha, PHI

REFERENCE BOOKS:

1. Lampson (Ed), Distributed Systems, Singer – Verlay NY 1981.
2. Mukesh Singhal, Niranjan G. Advanced Concepts in Operating Systems.

****

ADVANCED DBMS

Subject Code: 15CS614 **Credits: 03**
Hours/Week : 03 **Total Hours :39**

This Course will enable students to:

1. Outline the data structures used in the implementation of physical layer of a DBMS.

2. Write an optimized DBMS query so that the result will be faster.

3. Tell how the various relational operators are evaluated in a DBMS.

4. Compare the Relational DBMS with Object Databases and Distributed Databases.

5. Outline the security and authorization used in a DBMS.

UNIT-I

Overview of storage and indexing, disks and files:
Data on external storage; File organizations and indexing; Index data structures; Comparison of file organizations; Indexes and performance tuning. Memory hierarchy; RAID; Disk space management; Buffer manager; Files of records; Page formats and record formats.

Tree structured indexing: Intuition for tree indexes; Indexed Sequential Access Method (ISAM); B+ trees - Search, Insert, Delete, Duplicates; B+ trees in practice . **8 Hours**

UNIT-II

Hash based indexing: Static hashing; Extendible hashing, Linear hashing, comparisons.

External Sorting: When does a DBMS sort data? A simple two-way merge sort; External merge sort, Using B+ trees for sorting.

Evaluating Relational Operators: The Selection operation; General selection conditions; The Join operation; The Projection operation; The Set operations; Aggregate operations; The impact of buffering . **8 Hours**

UNIT-III

Query Optimization: Using Heuristics in Query Optimization, Using selectivity and cost estimates in Query Optimization, Overview of Query optimization in Oracle, Semantic Query Optimization .

Physical Database Design and Tuning: Introduction; Guidelines for index selection, examples; Clustering and indexing; Indexes that enable index-only plans; Overview of database tuning; Choices in tuning the conceptual schema; Choices in tuning queries and views; Impact of concurrency; DBMS benchmarking. **7 Hours**

UNIT-IV

Object Databases:

Concepts for Object Databases: Overview of Object-Oriented Concepts, Object Identity, Object Structure, and Type Constructors, Encapsulation of Operations, Methods, and Persistence, Type and Class Hierarchies and Inheritance, Complex Objects**;**

Object Database Standards, Languages, and Design: Overview of the Object Model of ODMG, The Object Definition Language ODL, The Object Query Language OQL, Overview of the C++ Language Binding, Object Database Conceptual Design**;**

Object-Relational and Extended-Relational Systems: Overview of SQL and its Object-Relational features, Object-Relational Features of Oracle 8 **. 8 Hours**

UNIT-V

Distributed Databases:

Distributed Database concepts; Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design; Types of Distributed Database Systems; Query Processing in Distributed Databases; Overview of Concurrency Control and Recovery in Distributed databases; Distributed databases in Oracle**.**

Security and Authorization:

Introduction to Database Security, Discretionary Access Control, Mandatory Access Control, Certifying Servers: The SSL Protocol, Digital Signatures, Role of the Database Administrator**.**

8 Hours

Course Outcomes

At the end of the course the student will be able to:

- **Analyze** the data structures used in the implementation of physical layer of a DBMS.
- **Write** an optimized DBMS query so that the result will be faster.
- **Describe** how the various relational operators are evaluated in a DBMS.
- **Compare** the Relational DBMS with Object Databases and Distributed Databases.
- **Analyze** the security and authorization used in a DBMS.

Mapping of POs & COs :

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H	H			H							
2	H	M			H				L		M	H
3	H	M			H				L		M	
4	H	H			H				L		M	
5	H	H			H						H	H

H : High M: Medium L : Low

TEXT BOOKS:

1. **Database Management Systems** – Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, McGraw-Hill, 2003.

2. **Fundamentals of Database Systems** – Elmasri and Navathe, 5th Edition, Addison-Wesley, 2007 .

REFERENCE BOOKS:

1. **Database Systems** – Connolly and Begg, 3rd Edition, Pearson Education, 2002.

2. **Distributed Databases principles and systems**, Ceri and Pelagatti, **McGraw-Hill, 2008**.

****

CLOUD COMPUTING AND INFRASTRUCTURE MANAGEMENT

Sub Code : 15CS615 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. **Outline** the fundamental ideas behind Cloud computing, and the evolution of the paradigm, its applicability; benefits as well as current and future challenges.
2. **Get** the basic idea and principles in Datacenter design and Management and find the importance of Virtualization in Cloud.
3. **Get** the idea of different Cloud deployment models and Cloud Delivery Models and their security issues.
4. **Tell how** Cloud Computing solves different problems in the present by considering different Cloud Vendors and their Cloud Design architecture.

UNIT – I

Eras of computing, Parallel vs. Distributed Computing, Elements of Parallel Computing- (What is parallel computing , hardware architecture for Parallel processing, approaches to parallel programming, levels of parallelism, Laws of caution). Elements of Distributed Computing- (General concepts and definitions, components of a distributed system, Architectural styles for distributed computing, models for inter-process communication, Technologies for distributed computing-Remote procedure call, Service oriented computing). **8 Hours**

UNIT – II

Classic data center, its elements, challenges and benefits. Data center management Steps in transitioning to cloud- consolidation, automation, IT as a service.

Cloud computing Architecture: - Introduction, Cloud reference models- (Architecture, Infrastructure/Hardware as a service, Platform as a service, Software as a service), Types of cloud – (Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds), Economics of cloud, Open challenges **6 Hours**

UNIT – III

Virtualization: – Introduction, characteristics of virtualized environments, taxonomy of virtualization technique- (execution of virtualization, other types of virtualization-Compute, Storage, Network, Desktop, Application). Virtualization and cloud computing, Pros and Cons of virtualization, Technology examples- XEN, VMware, Microsoft Hyper-V. **8 Hours**

UNIT – IV

Security Concerns, Risk Issues:- Cloud Computing- Security Concerns. A Closer Examination: Virtualization, A Closer Examination: Provisioning

Securing the Cloud: Key Strategies and Best Practices: - Overall Strategy: Effectively Managing Risk-Risk Management: Stages and Activities. Overview of Security Controls, Cloud Security Controls Must Meet Your Needs, NIST Definitions for Security Controls, Unclassified Models, Classified Model The Cloud Security Alliance Approach. The Limits of Security Controls - Security Exposure Will Vary over Time, Exploits Don't Play Fair. Best Practices: Best Practices for Cloud Computing- First Principals, Best Practices across the Cloud Community .Other Best Practices for Cloud Computing- Cloud Service Consumers, Cloud Service Providers. Security Monitoring-

The Purpose of Security Monitoring, Transforming an Event Stream, The Need for C.I.A. in Security Monitoring, the Opportunity for MaaS. **9 Hours**

UNIT - V

Case studies: Public cloud- AWS, Windows Azure, Google App Engine. Private Cloud- Open stack, Eucalyptus. **8 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. Define the concept of cloud computing business need and various networking methods.
2. Express the infrastructure management for cloud environment.
3. Practice the Virtualization at all levels using technology XEN, Vmware, Microsoft Hyper-v.
4. Explain the security concepts in cloud computing and securing the cloud.
5. Practice the case studies of public cloud such as AWS, Google App Engine and private cloud such as Open Stack.

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H	H										M
2		H									M	L
3		H									M	L
4		H									M	M
5	H										H	M

H : High M: Medium L : Low

TEXT BOOKS:

1. Buyya, Rajkumar, Christian Vecchiola and Thamarai Selvi. "Mastering Cloud Computing Fundamentals and Applications Programming." McGraw Hill, 2013.
2. G, Somasundarm and Alok Srivatsa. "Information Storage and Managemnt." EMC Education Services, Wiley Publishing Inc., 2009.
3. Sitaram, Dinakar and Geetha Manjunath. "Moving to the Cloud - Developing Apps in the World of Cloud Computing ." Elsevier, 2012.
4. Sosinsky, Barrie. "Cloud Computing Bible." Wiley India Pvt. Ltd , 2013.
5. Winkler, Vic(J.R). "Securing the Cloud - Cloud Computer Security Techniques and Tactics." Elsevier Inc, 2012.

REFERENCE BOOKS:

1. Hurwitz, Judith. "Cloud computing for dummies." Wiley India Pvt Ltd, 2011.
2. Rittinghouse, John. "Cloud computing – implementation, management and security." CRC Press, 1st edition, 2009.
3. Velte, Toby, Anthony Velte and Robert Elsenpete. "Cloud Computing, A Practical Approach." Tata McGraw-Hill Authors, 2010.

****

MANAGING BIG DATA

Sub Code : 15CS616 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. **Outline** the theory of big data ,and **explain** application of big data analytics
2. **Get** the idea of NoSQL databases, different types of NoSQL datastores and distribution models
3. **Analyze** the data with Hadoop and **design** of Hadoop distributed file system.
4. **Tells how** MapReduce programming model works,tesiting MRUnit and **get** the feeling of hadoop related like HBase and Hive

UNIT – I

UNDERSTANDING BIG DATA:

What is big data ,why big data , convergence of key trends , unstructured data , industry examples of big data, web analytics , big data and marketing ,fraud and big data , risk and big data, credit risk management ,big data and algorithmic trading , big data and healthcare ,big data in medicine, advertising and big data, big data technologies, introduction to Hadoop – open source technologies – cloud and big data **8 Hours**

UNIT – II

NOSQL DATA MANAGEMENT:

Introduction to NoSQL, Aggregate data models: aggregates, key-value and document data models,Relationships, graph databases, schemaless databases, Distribution models: sharding, master-slave replication, peer-peer replication, sharding and replication, MapReduce : partitioning and combining -- Composing Map-Reduce Calculations. **8 Hours**

UNIT - III

BASICS OF HADOOP:

Data format, Analyzing data with Hadoop, Scaling out, Hadoop streaming, Hadoop pipes. Design of Hadoop distributed file system (HDFS), HDFS concepts, Hadoop I/O, data integrity, compression, serialization **8 Hours**

UNIT – IV

MAPREDUCE APPLICATIONS:

MapReduce workflows, unit tests with MRUnit, test data and local tests, anatomy of MapReduce job run, job scheduling, shuffle and sort **7 Hours**

UNIT – V

HADOOP RELATED TOOLS:

Introduction to Hbase: The Dawn of Big Data, the Problem with Relational Database Systems, Introduction to Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, HiveQL queries. **8 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. **Describe** the role of big data ,application of big data anaytics and big data processing technology
2. **Identify** the concept NOSQL data management, its different types NoSQL data stores and distribution models.
3. **Analyse** the data with hadoop and **Design** of Hadoop distributed file system.
4. **Explain** the concept of MapReduce work flow
5. **Describe** the role Hbase and Hive in big data processing

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H		H	H	H			H				
2	H		H	L	H			H			H	H
3	H	H	H	L	H			H			H	H
4	H	H	H	L	H			H			H	H

H : High M: Medium L : Low

TEXT BOOKS:

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
2. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence",Copyright © 2013 Pearson Education, Inc. 2012.
3. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
4. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.

REFERENCE BOOKS:

1. Alex Holmes , “Hadoop in Practice “
2. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011
3. Alan Gates, "Programming Pig", O'Reilley, 2011.

****

INFORMATION AND STORAGE MANAGEMENT

Sub Code : 15CS617 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. Describe the concept of server centric architecture and storage centric architecture.
2. Compare different RAID levels, the techniques of instant copies, remote mirroring.
3. Differentiate Fibre channel SAN, NAS, iSCSI storage technologies.
4. Identify and outline various storage virtualization levels.
5. Indicate the application of storage networks, and identify the storage sharing.

UNIT - I

INTRODUCTION: Server Centric IT Architecture and its Limitations; Storage – Centric IT Architecture and its advantages; Case study: Replacing a server with Storage Networks;
INTELLIGENT DISK SUBSYSTEMS: Architecture of Intelligent Disk Subsystems; Hard disks and Internal I/O Channels,JBOD. **7 Hours**

UNIT – II

INTELLIGENT DISK SUBSYSTEMS: Storage virtualization using RAID, different RAID levels; Caching; Intelligent Disk Subsystems, Instant copies, Remote mirroring, LUN masking, Availability of Disk Subsystems
I/O TECHNIQUES: The Physical I/O path from the CPU to the Storage System; SCSI. Fiber Channel Protocol Stack; Fiber Channel SAN; **8 Hours**

UNIT – III

FILE SYSTEM AND NAS:
Local File Systems; Network file Systems and file servers; Shared Disk file systems; Comparison of fiber Channel SAN ,NAS AND iSCSI SAN.
STORAGE VIRTUALIZATION: Virtualization in the I/O Path; Limitations and requirements. **8 Hours**

UNIT – IV

STORAGE VIRTUALIZATION: Definition of Storage virtualization; Implementation Considerations; Storage virtualization on Block or file level; Storage virtualization on various levels of the storage Network; Symmetric and Asymmetric storage virtualization in the Network. **8 Hours**

UNIT – V

Application of storage networks: Networks in the I/O path , Data networks, voice networks and storage networks.
Storage Sharing: Disk storage pooling, Dynamic tape library sharing, Data sharing.
Availability of Data: Failure of an I/O bus , Failure of a server, Failure of a disk subsystem Failure of virtualization in the storage network Failure of a data centre based upon the case study 'protection of an important database'.**Adaptability and Scalability of IT Systems:** Clustering for load distribution, Web architecture, Web applications based upon the case study 'travel portal' **8 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. Identifythe Concept of server centric architecture, storage centric architecture, and intelligent disk subsystems architecture.
2. Compare different RAID levels, instant copies, remote mirroring, and I / o techniques
3. Compare fibre channel SAN, NAS, ISCSI storage technologies
4. Describe the various storage virtualization levels.
5. Locate the application of storage networks, and identifythe storage sharing.

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H		H		H							
2	H				M				L	M	M	
3	H				M				L		M	
4	H		H		H				L		M	
5	H		H		H						M	

H : High M: Medium L : Low

TEXT BOOK:

1. **Storage Networks Explained** – Ulf Troppens, Rainer Erkens and Wolfgang Muller, John Wiley & Sons, 2003.

REFERENCE BOOKS:

1. **Storage Networks: The Complete Reference** – Robert Spalding, Tata McGraw Hill, 2003.
2. **Storage Area Network Essentials: A Complete Guide to understanding and Implementing SANs** – Richard Barker and Paul Massiglia, John Wiley India, 2002.
3. **Storage Networking Fundamentals** Marc Farley, Cisco Press, 2005.

****

VLSI DESIGN

Sub Code : 15CS621 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours: 39**

Course Learning Objective:

This subject will enable the student to

1. Explain the VLSI and CMOS technology.
2. Design the circuits from the functional specifications.
3. Design the architecture of the chip.
4. Use Verilog and VHDL for design.

UNIT – I

Introduction : Digital Systems and VLSI: Why Design Integrated Circuits, Integrated Circuit manufacturing, CMOS Technology, Integrated Circuit Design Techniques, Transistors, Wires and Vias, Design Rules, Layout Design and Tools **8 Hours**

UNIT – II

LOGIC GATES, COMBINATIONAL LOGIC NETWORKS

Introduction, Combinational Logic Functions, Static Complementary gates, Wires and delay, switch logic, layout design methods, Simulation, Combinational Network Delay, Crosstalk, Power Optimization, Switch Logic Networks, Combinational Logic Testing **8 Hours**

UNIT – III

Sequential machines, Subsystem Design

Sequential Machines: Introduction , latches and flipflops, sequential systems and clocking disciplines, Sequential System Design, Power Optimization, Design Validation, Sequential Testing, Sub-system Design: Introduction, Subsystem Design Principles, Combinational Shifters adders, High-Density Memory, FPGAs, PLAs, **8 Hours**

UNIT – IV

Floor Planning, Architectural Design, Chip Design

Floorplanning; Introduction, Floorplanning Methods,Architectural Design: Introduction, Register-Transfer Design, High Level Synthesis, Architectures for low Power, Architecture Testing, Chip Design, Design methodology, Kitchen Timer Chip **7 Hours**

UNIT – V

CAD, Design Modeling

CAD systems and Algorithms: Introduction to CAD systems, Simulation, layout synthesis, Layout analysis, Timing Analysis and Optimization, Logic Synthesis, Test Generation, Design Modeling: Introduction: Hardware in VHDL **8 Hours**

Course Outcomes:

After studying this subject, the student should be able to:

1. Explain the fundamentals of digital VLSI design, design flow, CMOS technology.
2. Design and explain combinational and sequential circuits based on an abstract functional specification.
3. Prepare the floor planning and do the high level synthesis
4. Demonstrate knowledge and understanding of fundamental concepts in CAD.
5. Design and model digital circuits with Verilog HDL at behavioral, structural, and RTL Levels.

Mapping of POs & COs:

P Os COs	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
1	**H**	**M**										
2		**H**			**H**							
3											**H**	**L**
4			**M**		**H**						**H**	**H**
5	M				**M**						**H**	**H**

H : High M: Medium L : Low

TEXT BOOKS:

1. Wayne, Wolf, “Modern VLSI design: System on Silicon” Pearson Education, Second Edition
2. VLSI CAD – Niranjan N.Chiplunkar & Manjunath Kothari, PHI Learning, 2011

****

MULTICORE ARCHITECTURE & PROGRAMMING

Sub Code : 15CS622	**Credits : 03**
Hrs/Week : 3+0+0+0	**Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. Outline the principles of multi-core design and performance measurement
2. Illustrate the concept of parallelization and develop parallel programs
3. Identify the hurdles of parallelization and determine ways to handle these issues
4. Analyze the process of code optimization
5. Recognize the need and usage of multi threading tools

UNIT – I

INTRODUCTION TO MULTI-CORE ARCHITECTURE:
Introduction, Moore's law, Amdhal's law, Gustafson's law, Motivation for Multi-core processors, Types and levels of parallelism, Flynn's classification of multi-processors, Introduction to parallelization and vectorization: Data dependencies, SIMD technology, Hardware Multithreading vs. Software multi threading, Hyper threading, SMT, Case Study of multi-core processors: Intel, AMD, IBM/Sony. **8 Hours**

UNIT – II

CONCEPTS AND DESIGN OF PARALLEL AND THREAD PROGRAMMING:
Definition of thread and process, Parallel programming models, Parallel Programming constructs: Synchronization, Deadlock, Critical sections, Threading APIs- Win 32, POSIX threads**.** **7 Hours**

UNIT – III

PARALLEL PROGRAMMING:
MPI Model: Collective communication, Data decomposition, Communicators and topologies, point-to-point communication, MPI Library, OpenMP: Directives and clauses, environment variables, Programs using OpenMP and MPI. Introduction to intel TBB, Thread-Safeness, Cache related issues. **8 Hours**

UNIT – IV

MULTITHREADED PROGRAM DEBUGGING:
Benchmarks and other performance analysis tools, vTune Performance Analyzer, Thread Checker, Thread Profiler, hotspots, performance issues in algorithms, branch misprediction, cache organization, cache loads, efficiency, hardware and software prefetch. **7 Hours**

UNIT – V

COMPILER OPTIMIZATIONS AND PARALLEL ALGORITHMS:
Compilers for High performance Computing, compiler optimization, code and loop optimization, scalar and vector processing, temporal and spatial locality-matrix multiplication example. OS support to multi-core architectures. Parallel algorithms study and analysis - The Sieve of Eratosthenes, Floyd's algorithm, Matrix-Vector multiplication, Monte Carlo methods, Matrix Multiplication, Parallel Quicksort Algorithm. **9 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. Indentify the concept of multi-core architecture and motivation behind it.
2. Design parallel program using the multithreading concept.
3. Develop parallel programs using parallel programming frameworks.
4. Describe the concept of multithreaded program debugging.
5. Analyze the compiler optimizations and parallel algorithms.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H		H	H				L			H	
2	H	H			H	H		L		M	H	H
3	H	H	H		H	H		L		M	H	H
4	H	H			H			L		M	H	H
5	H	H	H	H	H	H		L	L	M	H	H

H : High M: Medium L : Low

TEXT BOOKS:

1. Multicore programming- Increasing performance through software multithreading,-- Shameem Akhter and Jason Roberts, Intel press
2. The software optimization cookbook- High performance Recipes for IA-32 Platforms – Richard Gerber, Aart J.C.Bik, Kevin B.Smith, Xinmin Tian, Intel press

REFERENCE BOOKS:

1. Advanced Compiler Design Implementation- Steven S.Muchnick, Morgan Kaufman Publishing 2000
2. www tutorials on introduction to parallel computing
3. www.openmp.org for OpenMP

****

MICROCONTROLLERS

Sub code : 15CS623 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

1. **Identify** basic building blocks of an embedded system.
2. **Compare** microcontroller and microprocessor architecture and functionality.
3. **Analyze** features of ARM Cortex Microcontroller.
4. **Use** ARM Kit STM32L-Discovery for simulating basic programs.

UNIT – I

Differences between Microprocessors and Microcontrollers, RISC and CISC CPU architectures, Harvard and Von-Neumann CPU architectures, Commercially available 8, 16 and 32 bit Microcontrollers from Intel, ARM and others.
Definition of Embedded system, Use of Microcontrollers in Embedded system, High level software development life cycle. ARM processor Architecture overview. **7 Hours**

UNIT – II

Introduction to ARMv7-M ISA and basic Programmer's model; ARM/Thumb assembly instructions, addressing modes, processor mode, register set, basic assembly instructions, MDK-ARM simulator, Assembly language programming. **7 Hours**

UNIT – III

C programming overview, I/O, stack, subroutine, logical and shift operation, KEIL utility, debugging in MDK-ARM simulator, Arithmetic operations, ARM prediction and condition execution, timers, pointers and advanced debugging, Finite State Machines, I/O synchronization, stack frames, FSMs in C. **7 Hours**

UNIT - IV

Cortex-M3 exception and interrupt handling, Systick timer and periodic interrupts, Memory mapped peripherals, LCD interface, MDK-ARM C programming, Mixing C and assembly, UART, SPI, I2C, CAN, DAC and ADC on ARM MCU, Review of threads and thread communication. **8 Hours**

UNIT - V

Hands on sessions to **carry out any 10 of the following exercises** using simulator and ARM Kit STM32L-Discovery:

a. Digital I/O, MCU pin direction, and logical functions, written in assembly and simulation
b. LEDs and switching, written in assembly & simulation
c. Use switches and LEDs, and control LED intensity using switches written in C and simulation

d. Traffic Light Controller with bits, written in assembly and simulation
e. Reset system using watchdog timer in case of error.
f. Simple FSM simulator in C
g. MCU bring-up and initialization, M3 vector table, setting up dummy handlers
h. LCD device driver and test, written in assembly and C (simulation and on board)
i. UART echo test (simulation and on board), and control LED intensity based on UART parameters (simulation and on board)
j. Display temperature on PC over UART (simulated and board)
k. Sample sound and plot amplitude vs. time on PC
l. Sample sound over microphone and display intensity through LEDs
m. FIFO queue, threads, mixture of assembly and C (simulated and board)
n. Real-time Position Monitor, ADC, interrupts, LCD, mixture of assembly and C (simulated and board)
o. Digital Piano or pacemaker using a DAC, C (simulated and board)
p. Generate RT clock using timers and output time over UART (simulation and board)

10 Hours

Course Outcomes:

After studying this subject, the student should be able to:

1. Identify the use/meaning of Memory, Central Processing Unit, Arithmetic and logical unit (ALU), Address Bus, Data Bus and Control lines of an embedded system.
2. Distinguish Microcontroller and Microprocessor design Architecture and the functionality.
3. Apply the C programming constructs in programming the devices.
4. Explain the architecture and programming of popular ARM Cortex Microcontroller
5. Design and Explain basic programs using simulator & ARM Kit STM32L-Discovery.

Mapping of Pos & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	**H**	**H**							**M**			
2	**H**	**H**										
3		**H**							**H**		**H**	**M**
4	H	**M**							**M**		**H**	**H**
5		**H**	**L**		**H**						**H**	**H**

H : High M: Medium L : Low

TEXT BOOKS AND RESOURCE BOOKS:

1. ARM Assembly Language: fundamentals and Techniques by William Hohi, CRC press, ISBN-10: 1439806101, ISBN-13: 978-1439806104

2. Definitive guide to ARM-Cortex M3, by Joseph Yiu, Newnes publisher, ISBN-10: 0750685344
   ISBN-13: 978-0750685344
3. www.arm.com/support/university/academic-resources.php

****

EMBEDDED AND REAL TIME SYSTEMS

Sub code : 15CS624 **Credits : 03**
Hrs/week : 3+0+0+0 **Total Hours : 39**

Prerequisite subjects: Computer Organization and Architecture (CS404)
Microprocessors and Peripherals (CS503) – Intel 8086 to Pentium
Operating Systems (CS505)

Course Learning Objectives:

1. **Explain** the concepts and principles of Embedded system design.
2. **Identify** basic building blocks of an embedded system .
3. **Assess** the benefits of Intel Atom based embedded system in terms of power consumption,
4. **Analyze** features of various RTOS.
5. **Use** Intel Atom boards in typical design of systems.
6. **Compare** various shared data handling techniques

UNIT – I

Embedded system definition, characteristics, design metrics; Processor, IC and design technologies; Embedded system examples, Digital Camera building blocks, Combinational and sequential building blocks. Use of DSP Processors, SoCs and Microcontrollers in embedded systems. Overview of 8051 microcontroller. **8 Hours**

UNIT – II

Timers, ADCs, Keypad controllers, LCD controllers, stepper motor and DC motor control, Custom Single Purpose processor design examples: GCD Generator, 4 bit multiplier, Communication bridge. Memory – Composing memory, memory hierarchy and Cache memory, interfacing-Serial, Parallel and Wireless Protocols. **7 Hours**

UNIT – III

Introduction to Real – Time Operating Systems, features, Examples of RTOS, typical RTOS functions. Interrupt handling and latency, Shared data problems, Tasks and Task States, Task scheduling, Inter-task communication and synchronization, Semaphores, Message Queues, Mailboxes and Pipes, Reentrant functions, Typical software architectures, Embedded Software development and testing tools, JTAG debugger, typical system boot flow diagram **9 Hours**

UNIT - IV

Intel ATOM Processor Architecture, Platform architecture and Micro architecture details, Overview of Assembly language programming of ATOM Processor, Low power issues of ATOM processor, ATOM processor series. **7 Hours**

UNIT – V

Intel ATOM Processor kit details, I/O options available, Keyboard and Mouse interface, GPS , GSM and RFID interface – **Hands On**, Overview of Device drivers. **8 Hours**

Course Outcomes:

After studying this subject, the student should be able to:

1. Identify basic building blocks of embedded systems.
2. Explain General purpose processor and the purpose of peripherals.
3. Illustrate the uses of RTOS.
4. Explain different features of real time operating systems.
5. Design an embedded system using Intel Atom boards.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H				L							
2	H				L							
3	H				L							
4	H		M	H	H				M	M	H	H

H : High M: Medium L : Low

TEXT BOOKS & RESOURCE BOOKS :

1. Frank Vahid and Tony Givargis, Embedded Systems Design – A unified Hardware/Software Introduction, John Wiley, 2002 (Chapter 1, 2, 4)
2. David E.Simon, *An Embedded Software Primer*, Pearson Education Asia, First Indian Reprint 2000. (Chapter 6,7, 8, 9)
3. Kenneth Ayala, 8051 Microcontroller Architecture, Programming and Applications, West publishing, 1991 (Selected chapters on Architecture of 8051)
4. Lori Matassa and Max Domeika " Break away with Intel Atom Processors: A guide to Architecture Migration" Intel Press, 2010 (Chapter 3, selected topics of Chapter 4 & 5)
5. Peter Barry, Patrik Crowley "Modern Embedded Computing", Morgn Kaufmann publishers, 2012. ISBN : 978-0-12-391490-3.
6. Intel Websites
7. NPTEL videos on Embedded Systems
8. Lab Manual of " Embedded system lab with ATOM Kit & Interfaces"

****

PROGRAMMING LANGUAGES

Sub Code : 15CS625	**Credits : 03**
Hrs/Week : 3+0+0+0	**Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. Outline the principles of scope of variables ,bindind.
2. Illustrate the concept of control flows.
3. Illustrate the pointers and recursive types and file input/output.
4. Analyze the object oriented principles and programming
5. Illustrate logic languages and scripting languages

UNIT – I

INTRODUCTION; NAMES, SCOPE, AND BINDINGS – 1 :
Language design; Programming language spectrum; Why study programming languages? Compilation and interpretation; Programming environments. Names, scope, and bindings: Concept of binding time; Object lifetime and storage management; Scope rules and implementing scope. The binding of reference environments; Binding within a scope; Separate compilation. **7 Hours**

UNIT – II

Control Flow : Expression evaluation, Structured and unstructured flow; Sequencing; Selection; Iteration; Recursion; DATA TYPES – 1: Type systems; Type checking; Records and variants; Arrays. **8 Hours**

UNIT – III

DATA TYPES - 2 :
Strings; Sets; Pointers and recursive types; Lists; Files, and Input/Output; Equality testing and assignment.
Subroutines and Control Abstraction - 1: Review of stack layout; Calling, sequences; Parameter passing; Generic subroutines and modules; Exception handling . **8 Hours**

UNIT – IV

CONTROL ABSTRACTION – 2; DATA ABSTRACTION, OBJECT ORIENTATION :
Control abstraction – 2: Coroutines. Data Abstraction, Object Orientation: Object oriented programming; Encapsulation and Inheritance; Multiple inheritance; **8 Hours**

UNIT – V

FUNCTIONAL LANGUAGES, LOGIC LANGUAGES, SCRIPTING LANGUAGES:
Functional Languages: Origins; Concepts; An overview of scheme; Evaluation order revisited; Higher-order functions; Functional programming in perspective. Logic Languages: Concepts; Prolog; Logic programming in perspective. Scripting Languages: Common characteristics.
8 Hours

Course Outcomes:

At the end of the course the student will be able to:

1. Recognize the concept of language design, compilation, interpretation, names, scope, and bindings .
2. Express the various control flow statements and different data types.
3. Represent various data types, subroutines, and control abstraction.
4. Describe the coroutines, data abstraction, and object orientation.
5. Explain the various functional languages, logic languages and scripting languages.

Mapping of POs & COs:

POs / COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H		H	H				L			H	
2		H			H	H			L		H	H
3	H							L			H	H
4		H			H				L		H	H
5		H	H					L			H	H

H : High M: Medium L : Low

TEXT BOOK:

1. **Programming Language Pragmatics** – Michael L. Scott, 2nd Edition, Elsevier, 2006.

REFERENCE BOOKS:

1. .**Programming Languages Concepts and Constructs** – Ravi Sethi, 2nd Edition, Pearson Education, 1996.
2. **Programming Languages** – Allen Tucker, Robert Nonan, Tata McGraw-Hill, 2002.

****

MULTICAST COMMUNICATIONS

Sub Code : 15CS626 **Credits : 03**
Hrs/Week : 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This Course will enable students to

1. Understand the preamble of group communications.
2. Comprehend and compare various routing algorithms in multicast communications.
3. Identify the switching technology for multicast communication in ATM Networks.
4. Discuss different transport protocols for multicast communications in internet.
5. Apply the concepts of multicast communications in Mbone Network.

UNIT – I

The basics of group communications: Types of communications; Multicast vs Unicast; Scalability; Applications of group communication; characteristics of groups; Special aspects of group communication. **7 Hours**

UNIT – II

Multicast Routing: Basic Routing algorithms; Group dynamics; scoping and multicast address allocation; Concepts of multicast routing; Multicast routing on the internet. **8 Hours**

UNIT – III

Multicast in ATM networks: The switching technology ATM; ATM multicast. **Transport protocols:** UDP; XTP **8 Hours**

UNIT - IV

Transport protocols: MTP; RMP; LBRM; SRM; RMTP **8 Hours**

UNIT – V

Mbone- The Multicast Backbone of the Internet: Mbonearchitecture ;Mbone applications; Mbone Tools; Outlook; Multicast Routing and Mobile Systems **8 Hours**

Course Outcomes:

At the end of the course the student will be able to:

1. Explain the basics of group communication and their applications.
2. Express the fundamental routing algorithms, concepts of multicast routing and multicast routing on internet.
3. Identify the switching technology ATM, ATM multicast, and transport protocols such as UDP / XTP
4. Describe the transport protocols MTP, RMP, LBRM, SRM, RMTP, and multicast backbone of the internet.

Mapping of POs & COs:

POs COs	a	b	c	d	e	f	g	h	i	j	k	l
1	H				H				M			
2			H					H				H
3			H									H
4	H							H				

H : High M: Medium L : Low

TEXT BOOK :

1. Ralph Wittmann and Martina Zittebart , Multicast Communication- protocols and Applications, Morgan Kaufmann Pub, 2001

****

DIGITAL SIGNAL PROCESSING

Sub code : 15CS627 **Credits : 03**
Hrs/Week: 3+0+0+0 **Total Hours : 39**

Course Learning Objectives:

This subject will enable the student to

1. Understand the behavior of discrete time systems in time& frequency domain.
2. Understand & Analyze the FFT algorithms.
3. Implement discrete time systems
4. Explain the features of TMS320c25 and TMS32067 processors.

UNIT – I

The Discrete Fourier Transform: Its Properties and Applications

Frequency Domain Sampling: The Discrete Fourier Transform: Frequency Domain Sampling and Reconstruction of Discrete-Time Signals, The Discrete Fourier Transform (DFT), The DFT as a Linear Transformation, Relationship of the DFT to other Transforms. Properties of the DFT: Periodicity, Linearity and Symmetry Properties, Multiplication of Two DFT's and Circular Convolution, Additional DFT Properties. Linear Filtering Methods Based on the DFT: Use of the DFT in Linear Filtering, Filtering of Long Data Sequences. Frequency Analysis of Signals using the DFT. **8 Hours**

UNIT – II

Efficient Computation of the DFT: Fast Fourier Transform Algs

Efficient Computation of the DFT: FFT Algorithms : Direct Computation of the DFT, Divide-and-Conquer Approach to Computation of the DFT, Radix-2 FFT Algorithms, Radix-4 FFT Algorithms, Split-Radix FFT Algorithms, Implementation of FFT Algorithms. Applications of FFT Algorithms: Efficient computation of the DFT of Two Real Sequences, Efficient computation of the DFT of a 2N-Point Real Sequence, Use of the FFT Algorithm in Linear filtering and Correlation. A Linear filtering approach to Computation of the DFT: The Goertzel Algorithm, The Chirp-Z Transform Algorithm. Quantization Effects in the Computation of the DFT: Quantization Errors in the Direct Computation of the DFT, Quantization Errors in FFT Algorithms. **8 Hours**

UNIT – III

Implementation of Discrete-Time Systems

Structures for the Realization of Discrete-Time Systems.
Structures for FIR Systems: Direct-Form Structures, Cascade-Form Structures, Frequency-Sampling Structures, Lattice Structure.
Structures for IIR Systems: Direct-Form Structures, Signal Flow Graphs and Transposed Structures, Cascade-Form Structures, Parallel-Form Structures, Lattice and Lattice-Ladder Structures for IIR Systems.
State-Space System Analysis and Structures: State-Space Descriptions of Systems Characterized by Difference Equations, Solution of the State Space Equations, Relationships between Input-Output and State-Space Descriptions, State-Space Analysis in the Z-Domain, Additional State Space Structures. **8 Hours**

UNIT – IV

Representation of Numbers

Fixed-Point Representation of Numbers, Binary Floating-Point Representation of Numbers, Errors Resulting from Rounding and Truncation.
Quantization of Filter Coefficients: Analysis of Sensuitivity to Quantizatior of Filter Coefficients, Quantization of Coefficients in FIR Filters.
Round-Off Effects in Digital Filters: Limit-Cycle Oscillations in Recursive Systems, Scaling to Prevent Overflow, Statistical Characterization of Quantization effects in Fixed-Point Realizations of Digital Filters.

8 Hours

UNIT – V

Digital Signal Processors

Architecture, features and instructions of Fixed and Floating point Processors. (TMS320c25 and TMS32067) **7 Hours**

Course Outcomes:

After studying this subject, the student should be able to:

1. Analyze the behavior of discrete-time systems in time & frequency domain.
2. Analyze and implement FFT algorithms.
3. Relate theoretical concepts to practical applications.
4. Summarize the working of TMS320c25 and TMS32067 processors.

Mapping of POs & COs:

P Os COs	a	b	c	d	e	f	g	h	i	j	k	l
1		**H**										
2		**H**										
3	**H**				**H**							
4				**H**					**M**		**H**	

H : High M: Medium L : Low

TEXT BOOKS :

1. Digital Signal Processing by John G. Proakis and Dimitris G. Manolakis, PHI, Third Edition 2003.
2. Signal Processing First by McLellan, Schafer & Yoder (Pearson) 2003.
3. Digital signal Processing with C and TMS 320c30, by Rulph Chassaing , John Wiley.

REFERENCE BOOKS:

1. Digital Signal Processing : System Analysis and Design by Paulo S. R. Diniz, Eduardo A. B. da Silva And Sergio L. Netto, Cambridge University Press, 2002.
2. Digital Signal Processing: A Computer Based Approach by Sanjit K. Mitra , Tata Mcgraw-Hill Edition 2001.
3. The Scientist and Engineers Guide to Digital Signal Processing –by Steven W. Smith 2nd Edition , 1999, California Technical Publishing.
4. Texas Instruments DSP Processors (320 family) data hand book.
5. DSP Processor Fundamentals Phil Lapsley, Jeff Bier, Amit Shoham and Edward A Lec, S.Chand, Delhi – 2000.
6. Digital Signal Processors: by B.Venkataramini & M.Bhaskar

****

EMPLOYABILITY SKILL DEVELOPMENT

Sub Code : 15IL001/002 **Credits : Nil (MLC)**
Hrs/Week : 1+0+0+0 **Total Hours : 12**

UNIT – I

Analytical Aptitude Skill: concept of analytical skill, definition-logical thinking and testing of Analytical Aptitude

UNIT – II

Quantitative Aptitude skill-Concept-definition-Preliminary requirement for development of quantitative skill- testing of quantitative skill.

UNIT – III

Verbal and ability skill – Knowledge and Vocabulary and grammer-comprehension-Verbal Reasoning skill

REFERENCE BOOKS:

1. Aggarwal R.S "Modern Approach to Logical Reasaning" S. Chanda Publication ,2008.
2. Aggarwal R.S "Quantitative Aptitude" S. Chand Publication ,2014.
3. Aggarwal R.S "Modern Approach to verbal and non verbal reasoning" S. Chanda Publication ,2013
4. Arun Sharma "Verbal ability and reading comprehension CAT" TMH Publications,2014
5. Ethnus Consultancy Pvt. Ltd " APTIMTRA: Your friend for cracking aptitude test", MGH Publications ,2014
6. Aggarwal R.S "Advanced objective general knowledge" S. Chanda Publication ,2014.

Examination pattern:

This course is a mandatory learning course without credit. Continuous internal examination (CIE) consists of 2 internal exams (20 marks each) and tasks (10 marks). There is no semester end examination (SEE). The student will be awarded PP or NP grade as per autonomous regulations.

****