                                                       4   0   3

Compiler Design

Course Objectives: To make the student to understand the process involved in a compiler, create an overall view of various types of translators, linkers, loaders, and phases of a compiler, understand what is syntax analysis, various types of parsers especially the top down approach, awareness among students the various types of bottom up parsers, understand the syntax analysis and, intermediate code generation, type checking, the role of symbol table and its organization, Code generation, machine independent code optimization and instruction scheduling.

Course Outcomes:
1. To introduce the major concept areas of language translation and compiler design
2. To develop an awareness of the function and complexity of compilers.
3. To provide practical, hands on experience in compiler design
4. Identify the similarities and differences among various parsing techniques and grammar transformation techniques

Unit–I:
Overview of language processing – pre-processors – compiler – assembler – interpreters, pre-processors, – linkers & loaders - structure of a compiler – phases of a compiler (TEXT BOOK 2). Lexical Analysis – Role of Lexical Analysis – Lexical Analysis Vs. Parsing – Token, patterns and Lexemes – Lexical Errors – Regular Expressions – Regular definitions for the language constructs – Strings, Sequences, Comments – Transition diagram for recognition of tokens, Reserved words and identifiers, Examples.

Unit–II
Syntax Analysis – discussion on CFG, LMD,RMD, parse trees, Role of a parser – classification of parsing techniques – Brute force approach, left recursion, left factoring, Top down parsing – First and Follow- LL(1) Grammars, Non-Recursive predictive parsing – Error recovery in predictive parsing.

Unit–III
What is bottom up parsing approach, Types of Bottom up approaches; Introduction to simple LR – Why LR Parsers – Model of an LR Parsers – Operator Precedence- Shift Reduce Parsing – Difference between LR and LL Parsers, Construction of SLR Tables.
More powerful LR parses, construction of CLR (1), LALR Parsing tables, Dangling ELSE Ambiguity, Error recovery in LR Parsing. Comparison of all bottoms up approaches with all top down approaches

Unit–IV
Semantic analysis, SDT Schemes, evaluation of semantic rules. Intermediate code, three address code, quadruples, triples, abstract syntax trees. Types and declarations, type Checking.

Unit–V
Symbol tables: use and need of symbol tables. Runtime Environment: storage organization, stack allocation, access to non-local data, heap management, parameter passing mechanisms, introduction to garbage collection. Reference counting garbage collectors.
Code generation: Issues, target language, Basic blocks & flow graphs, Simple code generator, Peephole optimization, Register allocation and assignment.

Unit–VI
Machine independent code optimization – semantic preserving transformations, global common sub expression elimination, copy propagation, dead code elimination, constant folding, strength reduction, loop optimization. Instruction scheduling, inter procedural optimization.

TEXT BOOKS:

1. Compilers, Principles Techniques and Tools- Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D. Ullman,2nd ed, Pearson,2007.
2. Compiler Design, K. Muneeswaran, Oxford.

REFERENCE BOOKS:

1. Engineering a compiler, 2nd edition, Keith D.Cooper & Linda Torczon, Morgan Kaufman.
2. http://www.nptel.iitm.ac.in/downloads/106108052/
3. Principles of compiler design, V. Raghavan, 2nd ed, TMH, 2011.
4. Compiler construction, Principles and Practice, Kenneth C Louden, CENGAGE
5. Implementations of Compiler, A new approach to Compilers including the algebraic methods, Yunlinsu, SPRINGER

1. Engineering a compiler, 2nd edition, Keith D.Cooper & Linda Torczon, Morgan Kaufman.
2. http://www.nptel.iitm.ac.in/downloads/106108052/
3. Principles of compiler design, V. Raghavan, 2nd ed, TMH, 2011.
4. Compiler construction, Principles and Practice, Kenneth C Louden, CENGAGE
5. Implementations of Compiler, A new approach to Compilers including the algebraic methods, Yunlinsu, SPRINGER

	T	P	C
III Year – I SEMESTER	4	0	3

Data Communication

Course Objectives:
1. To have a detailed study of various analog and digital modulation and demodulation techniques
2. To have a thorough knowledge of various multiplexing schemes and Data communication protocols
3. To know about the standards and mechanisms of television systems

Course Outcomes:
1. Knowledge of working of basic communication systems
2. Ability to evaluate alternative models of communication system design

Syllabus:

Unit I:
INTRODUCTION TO DATA COMMUNICATIONS AND NETWORKING: Standards Organizations for Data Communications, Layered Network Architecture, Open Systems Interconnection, Data Communications Circuits, Serial and parallel Data Transmission, Data communications Networks, Alternate Protocol Suites.
SIGNALS, NOISE, MODULATION, AND DEMODULATION:Signal Analysis, Electrical Noise and Signal-to-Noise Ratio, Analog Modulation Systems, Information Capacity, Bits, Bit Rate, Baud, and *M*-ary Encoding, Digital Modulation.

Unit II :
METALLIC CABLE TRANSMISSION MEDIA: Metallic Transmission Lines, Transverse Electromagnetic Waves, Characteristics of Electromagnetic Waves
OPTICAL FIBER TRANSMISSION MEDIA: Advantages of Optical Fiber cables, Disadvantages of Optical Fiber Cables, Electromagnetic spectrum, Optical Fiber Communications System Block Diagram, Optical Fiber construction, Propagation of Light Through an Optical fiber Cable, Optical Fiber Modes and Classifications, Optical Fiber Comparison, Losses in Optical Fiber Cables, Light sources, Light Detectors, Lasers.

Unit III :
DIGITAL TRANSMISSION: Pulse Modulation, Pulse code Modulation, Dynamic Range, Signal Voltage –to-Quantization Noise Voltage Ratio, Linear Versus Nonlinear PCM Codes, Companding, PCM Line Speed, Delta Modulation PCM and Differential PCM.
MULTIPLEXING AND T CARRIERS: Time- Division Multiplexing, T1 Digital Carrier System, Digital Line Encoding, T Carrier systems, Frequency- Division Multiplexing, Wavelength- Division Multiplexing, Synchronous Optical Network

Unit IV:
WIRELESS COMMUNICATIONS SYSTEMS: Electromagnetic Polarization, Electromagnetic Radiation, Optical Properties of Radio Waves, Terrestrial Propagation of Electromagnetic Waves, Skip Distance, Free-Space Path Loss, Microwave Communications Systems, Satellite Communications Systems.

Unit V:
TELEPHONE INSTRUMENTS AND SIGNALS: The Subscriber Loop, Standard Telephone Set, Basic Telephone Call Procedures, Call Progress Tones and Signals, Cordless Telephones, Caller ID, Electronic Telephones, Paging systems.
CELLULAR TELEPHONE SYSTEMS: First- Generation Analog Cellular Telephone, Personal Communications system, Second-Generation Cellular Telephone Systems, N-AMPS, Digital Cellular Telephone, Interim Standard, Global system for Mobile Communications.

Unit VI:
DATA COMMUNICATIONS CODES, ERROR CONTROL, AND DATA FORMATS:
Data Communications Character Codes, Bar Codes, Error Control, Error Detection and Correction, Character Synchronization.
DATA COMMUNICATIONS EQUIPMENT: Digital Service Unit and Channel Service Unit, Voice- Band Data Communication Modems, Bell Systems-Compatible Voice- Band Modems, Voice- Band Modem Block Diagram, Voice- Band Modem Classifications, Asynchronous Voice-Band Modems, Synchronous Voice-Band Modems, Modem Synchronization, 56K Modems, Modem Control: The AT Command Set, Cable Modems.

TEXT BOOKS:

1. Introduction to Data Communications and Networking, Wayne Tomasi, Pearson Education.

Reference Books :

1. Data Communications and Networking, Behrouz A Forouzan, Fourth Edition.TMH.
2. Data and Computer communications, 8/e, William Stallings, PHI.
3. Computer Communications and Networking Technologies, Gallow, Second Edition Thomson
4. Computer Networking and Internet, Fred Halsll, Lingana Gouda Kulkarni, Fifth Edition, Pearson Education

T	P	C
4	0	3

Principles of Programming Languages

Course objectives:
1. To understand and describe syntax and semantics of programming languages
2. To understand data, data types, and basic statements
3. To understand call-return architecture and ways of implementing them
4. To understand object-orientation, concurrency, and event handling in programming languages
5. To develop programs in non-procedural programming paradigms

Course Outcomes:
Upon Completion of the course, the students will be able to
1. Describe syntax and semantics of programming languages
2. Explain data, data types, and basic statements of programming languages
3. Design and implement subprogram constructs, Apply object - oriented, concurrency, and event handling programming constructs
4. Develop programs in Scheme, ML, and Prolog
5. Understand and adopt new programming languages

Syllabus:

UNIT I :
SYNTAX AND SEMANTICS: Evolution of programming languages, describing syntax, context,free grammars, attribute grammars, describing semantics, lexical analysis, parsing, recursive - decent bottom - up parsing

UNIT II:
DATA, DATA TYPES, AND BASIC STATEMENTS: Names, variables, binding, type checking, scope, scope rules, lifetime and garbage collection, primitive data types, strings, array types, associative arrays, record types, union types, pointers and references, Arithmetic expressions, overloaded operators, type conversions, relational and boolean expressions , assignment statements , mixed mode assignments, control structures – selection, iterations, branching, guarded Statements

UNIT III:
SUBPROGRAMS AND IMPLEMENTATIONS: Subprograms, design issues, local referencing, parameter passing, overloaded methods, generic methods, design issues for functions, semantics of call and return, implementing simple subprograms, stack and dynamic local variables, nested subprograms, blocks, dynamic scoping

UNIT IV:
OBJECT- ORIENTATION, CONCURRENCY, AND EVENT HANDLING: Object – orientation, design issues for OOP languages, implementation of object, oriented constructs, concurrency, semaphores, Monitors, message passing, threads, statement level concurrency, exception handling, event handling

UNIT V :
FUNCTIONAL PROGRAMMING LANGUAGES: Introduction to lambda calculus, fundamentals of functional programming languages, Programming with Scheme, –
Programming with ML,

UNIT VI :
LOGIC PROGRAMMING LANGUAGES: Introduction to logic and logic programming, –
Programming with Prolog, multi - paradigm languages

TEXT BOOKS:
1. Robert W. Sebesta, "Concepts of Programming Languages", Tenth Edition, Addison Wesley, 2012.
2. Programming Langugaes, Principles & Paradigms, 2ed, Allen B Tucker, Robert E Noonan, TMH

REFERENCES:
1. R. Kent Dybvig, "The Scheme programming language", Fourth Edition, MIT Press, 2009.
2. Jeffrey D. Ullman, "Elements of ML programming", Second Edition, Prentice Hall, 1998.
3. Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009.
4. W. F. Clocksin and C. S. Mellish, "Programming in Prolog: Using the ISO Standard", Fifth Edition, Springer, 2003

Database Management Systems

Course Objectives:
Provides students with theoretical knowledge and practical skills in the use of databases and database management systems in information technology applications. The logical design, physical design and implementation of relational databases are covered.

Course Outcomes:
- define a Database Management System
- give a description of the Database Management structure
- understand the applications of Databases
- know the advantages and disadvantages of the different models
- compare relational model with the Structured Query Language (SQL)
- know the constraints and controversies associated with relational database model.
- know the rules guiding transaction ACID
- understand the concept of data planning and Database design
- identify the various functions of Database Administrator

Syllabus:

Unit – I: INTRODUCTION
Database system, Characteristics (Database Vs File System), Database Users(Actors on Scene, Workers behind the scene), Advantages of Data base systems, Database applications.
Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Unit – II:
RELATIONAL MODEL : Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance
BASIC SQL : Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).

Unit – III:
Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.
SQL : Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view(updatable and non-updatable), relational set operations.

Unit – IV:
SCHEMA REFINEMENT (NORMALIZATION) : Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency(1NF, 2NF and 3 NF), concept of surrogate key, Boyce-codd normal form(BCNF), Lossless join and dependency preserving decomposition, Fourth normal form(4NF).

Unit – V:
TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL : Transaction, properties of transactions, transaction log, and transaction management with SQL using commit rollback and savepoint.
Concurrency control for lost updates, uncommitted data, inconsistent retrievals and the Scheduler. Concurrency control with locking methods : lock granularity, lock types, two phase locking for ensuring serializability, deadlocks, Concurrency control with time stamp ordering : Wait/Die and Wound/Wait Schemes, Database Recovery management : Transaction recovery.
SQL constructs that grant access or revoke access from user or user groups. Basic PL/SQL procedures, functions and triggers.

UNIT – VI:

STORAGE AND INDEXING : Database file organization, file organization on disk, heap files and sorted files, hashing, single and multi-level indexes, dynamic multilevel indexing using B-Tree and B+ tree, index on multiple keys.

Text Books :
1. Database Management Systems, 3/e Raghuram Krishnan, Johannes Gehrke, TMH
2. Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA
3. Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Reference Books :
1. Database System Concepts. 5/e Silberschatz, Korth, TMH
2. Introduction to Database Systems, 8/e C J Date, PEA
3. The Database book principles & practice using Oracle/MySql Narain Gehani, University Press.

Operating Systems

Course Objectives:

To gain knowledge about the Operating Systems concepts such as process, main memory management, secondary memory management, CPU and disk scheduling etc

Course Outcomes:

By the end of the course student will be able to

- describe the general architecture of computers
- describe, contrast and compare differing structures for operating Systems
- understand and analyse theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files

Syllabus:

UNIT-I:

Computer System and Operating System Overview: Overview of computer operating systems, operating systems functions, protection and security, distributed systems, special purpose systems, operating systems structures and systems calls, operating systems generation.

UNIT-II:

Process Management – Process concept- process scheduling, operations, Inter process communication. Multi Thread programming models. Process scheduling criteria and algorithms, and their evaluation.

UNIT-III:

Concurrency: Process synchronization, the critical- section problem, Peterson's Solution, synchronization Hardware, semaphores, classic problems of synchronization, monitors, Synchronization examples

UNIT-IV:

Memory Management: Swapping, contiguous memory allocation, paging, structure of the page table, segmentation
Virtual Memory Management:
virtual memory, demand paging, page-Replacement, algorithms, Allocation of Frames, Thrashing

UNIT-V:

Principles of deadlock – system model, deadlock characterization, deadlock prevention, detection and avoidance, recovery form deadlock,

UNIT-VI:

File system Interface- the concept of a file, Access Methods, Directory structure, File system mounting, file sharing, protection.
File System implementation- File system structure, allocation methods, free-space management
Mass-storage structure overview of Mass-storage structure, Disk structure, disk attachment, disk scheduling

TEXT BOOKS:

1. Operating System Concepts- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th   Edition, John Wiley.
2. Operating Systems' – Internal and Design Principles Stallings, Sixth Edition–2005,   Pearson education

REFERENCE BOOKS:

1. http://nptel.iitm.ac.in/courses/Webcourse-contents/IISc-BANG/ Operating%20Systems/New_index1.html
2. Operating systems- A Concept based Approach-D.M.Dhamdhere, 2nd Edition, TMH
3. Operating System A Design Approach-Crowley, TMH.
4. Modern Operating Systems, Andrew S Tanenbaum 3rd edition PHI.

Compiler Design Lab

Course Objectives:
To enlighten the student with knowledge base in compiler design and its applications

Course Outcomes:
Demonstrate a working understanding of the process of lexical analysis, parsing and other compiler design aspects.

Lab Experiments:

1. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines
2. Simulate First and Follow of a Grammar.
3. Develop an operator precedence parser for a given language.
4. Construct a recursive descent parser for an expression.
5. Construct a LL(1) parser for an expression
6. Design predictive parser for the given language
7. Implementation of shift reduce parsing algorithm.
8. Design a LALR bottom up parser for the given language.
9. Implement the lexical analyzer using JLex, flex or lex or other lexical analyzer generating tools
10. Write a program to perform loop unrolling.
11. Convert the BNF rules into YACC form and write code to generate abstract syntax tree.
12. Write a program for constant propagation.

Operating System Lab

Objective :
- To provide an understanding of the design aspects of operating system

Recommended Systems/Software Requirements:
- Intel based desktop PC with minimum of 166 MHZ or faster processor with atleast 64 MB RAM and 100 MB free disk space

Lab Experiments:

1. Simulate the following CPU scheduling algorithms
   a) Round Robin   b) SJF   c) FCFS   d) Priority

2. Loading executable programs into memory and execute System Call implementation-read(), write(), open () and close()

3. . Multiprogramming-Memory management- Implementation of Fork(), Wait(), Exec() and Exit() System calls

4. Simulate all File allocation strategies   a) Sequenced   b) Indexedc) Linked

5. Simulate MVT and MFT

6. Simulate all File Organization Techniques
    a) Single level directory   b) Two level   c) Hierarchical   d) DAG

7. Simulate Bankers Algorithm for Dead Lock Avoidance

8. Simulate Bankers Algorithm for Dead Lock Prevention.

9. Simulate all page replacement algorithms.
    a) FIFO   b) LRU   c) LFU   etc….

10. Simulate Paging Technique of memory management.

Database Management Systems Lab

Objectives:
· To teach the student database design and query and PL/SQL.

System/Software Requirements:
· Intel based desktop PC
· Mysql /Oracle latest version Recommended

PROGRAMS LIST:

1) Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

2) Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.
   Example:- Select the roll number and name of the student who secured fourth rank in   the class.

3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and
   MIN), GROUP BY, HAVING and Creation and dropping of Views.

4) Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

5)
   i)Creation of simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)

   ii)Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.

8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.

10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.

11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.

12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

TEXT BOOKS :

1) ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson Education 3rd Edition
2) ORACLE DATA BASE LOG PL/SQL Programming SCOTT URMAN, Tata Mc-Graw Hill.

3) SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.

4) Data Base Management System, Oracle SQL and PL/SQL, Pranab kumar Das Gupta, P Radha Krishna, PHI

Linux Programming Lab

Objectives:
To give a practical orientation of programming in Linux environment using system calls and advanced concepts in unix programming

PROGRAMS LIST:

1. Write C programs that uses open, read, write system calls.
2. Write C programs that differentiates FILE *(file stream pointers in C standard library) and file descriptors by using functions such as fdopen, fileno.
3. Write a C program which displays a given files meta data by using stat system call and st_mode structure.
4. Write a C program which lists all the files of current working directory whose size is more than given number of data blocks.
5. Write a C program which lists all the files of current working directory which contains hard link files.
6. Write a C program to emulates file system checking utility (fsck command) using system calls.
7. Example C program which supports that child process inherits environment variables, command line arguments, opened' files.
8. Simple C programs to have process trees and process chains.
9. Simple C program that demonstrates the failure of fork system call because of crossing system limits.
10. Simple C programs to demonstrate the use of pipe system call for inter process communication and also emulating piping in shell.
11. Simple C programs to demonstrate the use of popen standard library function call for inter process communication and also emulating piping in shell.
12. Simple C program to use named pipes for inter process communication.
13. Simple C programs to illustrate the use of exec family of functions.
14. Write a C program which emulates simple shell.
15. Write C program to create a thread using pthreads library and let it run its function.
16. Write a C program to illustrate concurrent execution of threads using pthreads library.
17. Write a C program to simulate ptrhead_create function failure by repeatedly calling the same.
18. Write a C program which creates a thread using pthread and passes arguments to the thread function.
19. Write C programs which uses sigset, sifillset, sigprocmask, related system calls and structures.
20. Write a C program to simulate memory segment violation run time error and implement a signal handler (both reliable and unreliable) which handles situation.
21. Write a C program to illustrate the use of sbrk system call.
22. Write a C program to illustrate inter process communication via message queues.
23. Write a C program to illustrate inter process communication via shared memory.
24. Write a C program to simulate producer and consumer problem using semaphores, shared memory, and fork.
25. Write a C program to simulate producer and consumer problem using semaphores, shared memory, and pthread_create.
26. Write a C program to simulate producer and consumer problem using muexes, shared memory, and threads.
27. Write socket Programs in C for Echo/Ping/Talk Commands.
28. Create a Socket (TCP) between two computers and enable file transfer between them.
29. Write a Program to implement Remote Command Execution.
30. Write a code simulating ARP/RARP.

INTELLECTUAL PROPERTY RIGHTS AND PATENTS – 1

Unit 1

Introduction to Intellectual Property Law – The Evolutionary Past - The IPR Tool Kit- Para -Legal Tasks in Intellectual Property Law – Ethical

obligations in Para Legal Tasks in Intellectual Property Law - Introduction to Cyber Law
– Innovations and Inventions Trade related Intellectual Property Right

Unit 2

Introduction to Trade mark – Trade mark Registration Process – Post registration procedures – Trade mark maintenance - Transfer of Rights - Inter partes Proceeding – Infringement - Dilution Ownership of Trade mark
– Likelihood of confusion - Trademarks claims – Trade marks Litigations – International Trade mark Law –

Unit 3

Introduction to Copyrights – – Principles of Copyright Principles -The subjects Matter of Copy right – The Rights Afforded by Copyright Law – Copy right Ownership, Transfer and duration – Right to prepare Derivative works – Rights of Distribution – Rights of Perform the work Publicity Copyright Formalities and Registrations - Limitions - Copyright disputes and International Copyright Law – Semiconductor Chip Protection Act

Unit 4

Introduction to Trade Secret – Maintaining Trade Secret – Physical Secrurity –Employee Limitation - Employee confidentiality agreement - Trade Secret Law - Unfair Competation – Trade Secret Letigation – Breach of Contract – Applying State Law

Books:

1. Deborah E.Bouchoux: "Intellectual Property". Cengage learning , New Delhi
2. Kompal Bansal & Parishit Bansal "Fundamentals of IPR for Engineers", BS Publications (Press)
3. Cyber Law. Texts & Cases, South-Western's Special Topics Collections

4. Prabhuddha Ganguli: ' Intellectual Property Rights" Tata Mc-Graw – Hill, New Delhi
5. Richard Stim: "Intellectual Property", Cengage Learning, New Delhi.
6. R. Radha Krishnan, S. Balasubramanian: "Intellectual Property Rights", Excel Books. New Delhi.

M.Ashok Kumar and Mohd.Iqbal Ali: "Intellectual Property Right" Serials Pub.

III Year – I SEMESTER	**T**	**P**	**C**
	0	**0**	**1**

Seminar