III SEMESTER
SYSTEMS SOFTWARE

Subject Code	**: 07MCA31**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Machine Architecture:
Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) - SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples.

6 Hours

Assemblers -I:
Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures, Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation.

6 Hours

Assemblers -II:
Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler, Implementation Examples - MASM Assembler.

6 Hours

Loaders and Linkers:
Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features – Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage, Bootstrap Loaders, Implementation Examples - MS-DOS Linker.

8 Hours

Editors and Debugging Systems:
Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship with other parts of the System, User-Interface Criteria

6 Hours

Macro Processor:

Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine-Independent Macro Processor Features - Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor Design Options - Recursive Macro Expansion, General-Purpose Macro Processors, Macro Processing within Language Translators, Implementation Examples - MASM Macro Processor, ANSI C Macro Processor.

8 Hours

Lex and Yacc – 1
Lex and Yacc - The Simplest Lex Program, Recognizing Words With Lex, Symbol Tables, Grammars, Parser-Lexer Communication, The Parts of Speech Lexer, A Yacc Parser, The Rules Section, Running LEX and YACC, Lex and Hand- Written Lexers, Using LEX - Regular Expression, Examples of Regular Expressions, A Word Counting Program, Parsing a Command Line.

6 Hours

Lex and Yacc - 2
Using Yacc – Grammars, Recursive Rules, Shift/Reduce Parsing, What Yacc Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, Symbol Values and Actions, The LEXER, Compiling and Running a Simple Parser, Arithmetic expressions and ambiguity, Variables and typed tokens.

6 Hours

TEXT BOOKS:

1. Leland. L.Beck: **System Software**, 3rd Edition, Addison-Wesley, 1997. (Chapters 1.1 to 1.3, 2 (except 2.5.2 and 2.5.3), 3 (except 3.5.2 and 3.5.3), 4 (except 4.4.3))
2. John.R.Levine, Tony Mason and Doug Brown: Lex and Yacc, O'Reilly, SPD, 1998. (Chapters 1, 2 (Page 2-42), 3 (Page 51-65))

REFERENCE BOOKS:

1. D.M.Dhamdhere: **System Programming and Operating Systems**, 2nd Edition, Tata McGraw - Hill, 1999.

COMPUTER NETWORKS

Subject Code	**: 07MCA32**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03

Total Hours	: 52	Exam Marks	100

Foundation:
Building a Network; Applications; Requirements; Network Architecture; Implementing Network software; Performance.

8 Hours

Direct Link Networks:
Physically connecting hosts; Hardware building blocks; Encoding; Framing; Error detection; Reliable transmission; Ethernet (802.3); Ring; (802.5, FDDI, 802.17); Wireless (802.15.1, 802.11, 802.16, Cell Phone Technologies).

12 Hours

Packet Switching:
Switching and forwarding; Bridges and LAN Switches

7 Hours

Internetworking:
Simple internetworking (IP); Routing; Global Internet.

12 Hours

End –to-End Protocols; Resource Allocation Issues:
Simple demultiplexer (UDP); Reliable byte stream (TCP); Issues in resource allocation

7 Hours

Applications:
Application-Layer overview; Domain Name System; Remote Login protocols; Electronic mail; File transfer and FTP; World Wide Web and HTTP; Overview of VoIP telephony; VoIP signaling protocols.

6 Hours

TEXT BOOKS:

1. Larry L. Peterson and Bruce S. David: **Computer Networks – A Systems Approach**, 4th Edition, Elsevier, 2007.
(Chapters 1, 2, 3.1, 3.2, 4.1, 4.2, 4.3, 5.1, 5.2, 6.1)
2. Nader F. Mir: **Computer and Communication Networks**, Pearson Education, 2007.
(Chapters 9.1 to 9.6, 18.1, 18.2)

REFERENCE BOOKS:

1. Behrouz A. Forouzan: **Data Communications and Networking**, 4th Edition, Tata McGraw-Hill, 2006.
2. William Stallings: **Data and Computer Communication**, 8th Edition, Pearson Education / PHI, 2007.
3. Alberto Leon-Garcia and Indra Widjaja: **Communication Networks** -Fundamental Concepts and Key architectures, 2nd Edition Tata McGraw-Hill, 2004.

OBJECT ORIENTED PROGRAMMING WITH C++

Subject Code	**: 07MCA33**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Overview of OOP:
Object Oriented paradigm, Structured vs. Object Oriented Paradigm. Elements of Object Oriented Programming, Object, Classes, Encapsulation & data abstraction, Inheritance, Polymorphism etc.

3 Hours

C++ Overview:
Introduction, different data types, operators, expressions, qualifiers, arrays and strings.

4 Hours

Modular Programming with Functions:
Function Components, argument passing, inline functions, function overriding, function overloading, function templates, recursive functions.

8 Hours

Classes & Objects:
Introduction, Class Specification, Class Objects, access members, defining member functions, data hiding, constructors, destructors, parameterized constructors, static data members, functions, scope resolution operator, Passing objects as arguments, returning objects, friend functions & classes, arrays of objects, Dynamic objects – Pointers to objects, Class members, Operator overloading using friend functions such as ++, --, [ ] etc.

18 Hours

Inheritance Virtual functions and Polymorphism:
Base Class, Inheritance & protected members, protected base class inheritance, Inheriting multiple base classes, Constructors, Destructors & Inheritance, Passing parameters to base Class Constructors, Granting access, Virtual base classes. Virtual functions – Calling a Virtual function through a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, pure virtual functions, abstract classes, using Virtual functions, Early & late binding.

12 Hours

Generic Programming with Templates

Introduction, Function templates, Class templates.

3 Hours

I/O

IO Stream Library, output operator <<, input >>, additional i/o operators, overloading the output operator <<, overloading the io operator >>, file input & output.

4 Hours

TEXT BOOKS:

1. H. Schild: **C++ The Complete Reference**, 4th Edition, Tata McGraw Hill, 2003.
   (Chapters 11 to 17, 20, 21)
2. K R Venugopal, Rajkumar, T., Ravi Shankar: **Mastering C++**, Tata McGraw Hill, 1997.
   (Chapter 1, 3 to 18)

REFERENCE BOOK:

1. Stanley B.Lippmann, Josee Lajore: **C++Primer,** 4th d Edition, Addison Wesley, 2005.

DATABASE MANAGEMENT SYSTEMS

Subject Code	**: 07MCA34**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction:

Introduction; Characteristics of Database approach; Actors on the screen; Workers behind the scene; Advantages of using DBMS approach; A brief history of database applications; when not to use a DBMS. Data models, schemas and instances; Three-schema architecture and data independence; Database languages and interfaces; the database system environment; Centralized and client-server architectures; Classification of Database Management systems.

7 Hours

Entity-Relationship (ER) 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; ER Diagrams, Naming Conventions and Design Issues; Relationship types of degree higher than two.

7 Hours

Relational Model and Relational Algebra:

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions 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.

8 Hours

SQL :

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; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL; Additional features of SQL; Database programming issues and techniques; Embedded SQL, Dynamic SQL; Database stored procedures and SQL / PSM.

16 Hours

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.

8 Hours

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.

6 Hours

TEXT BOOKS:

1. Elmasri and Navathe: **Fundamentals of Database ystems**, 5th Edition, Addison-Wesley, 2007
   (Chapters 1, 2, 3 except 3.8, 5, 6.1 to 6.5, 7.1, 8, 9.1, 9.2 except SQLJ, 9.4, 10)
2. Raghu Ramakrishnan and Johannes Gehrke: **Database Management Systems**, 3rd Edition, McGraw-Hill, 2003.
   (Chapters 16, 17.1, 17.2, 18)

REFERENCE BOOKS:

1. Silberschatz, Korth and Sudharshan: **Data base System Concepts**, 5th Edition, Mc-GrawHill, 2006.
2. C.J. Date, A. Kannan, S. Swamynatham: **A Introduction to Database Systems**, 8th Edition, Pearson education, 2006.

OPERATING SYSTEMS

Subject Code	**: 07MCA35**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction to Operating Systems, System structures:

What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Distributed system; Special-purpose systems; Computing environments. 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.

6 Hours

Process Management:

Process concept; Process scheduling; Operations on processes; Inter-process communication. Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; Thread scheduling.

7 Hours

Process Synchronization:

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

7 Hours

Deadlocks:

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

6 Hours

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.

7 Hours

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.

7 Hours

Secondary Storage Structures,Protection

Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, access matrix , Implementation of access matrix, Access control, Revocation of access rights, Capability-Based systems.

6 Hours

Case Study: The Linux Operating System:

Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory management; File systems, Input and output; Inter-process communication.

6 Hours

TEXT BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin , Greg Gagne: **Operating System Principles**, 7th edition, Wiley-India, 2006.
   (Chapters: 1, 2, 3.1 to 3.4 , 4.1 to 4.4, 5.1 to 5.5, 6.1 to 6.7, 7, 8.1 to 8.6, 9.1 to 9.6, 10, 11.1 to 11.5, 12.1 to 12.6, 17.1 to 17.8, 21.1 to 21.9)

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.

SYSTEM PROGRAMMING LABORATORY

Subject Code	**: 07MCA36**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

PART- A

Execute of the following programs using LEX:

1) Program to count the number of vowels and consonants in a given string.
2) Program to count the number of characters, words, spaces and lines in a given input file.
3) Program to count number of
   a) Positive and negative integers
   b) Positive and negative fractions

4) Program to count the numbers of comment lines in a given C program. Also eliminate them and copy that program into separate file.
5) Program to count the number of ‘scanf’ and ‘printf’ statements in a C program. Replace them with ‘readf’ and ‘writef’ statements respectively.
6) Program to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately.
7) Program to recognize and count the number of identifiers in a given input file.

PART- B

Execute of the following programs using YACC:

1) Program to test the validity of a simple expression involving operators +, -, * and /.
2) Program to recognize nested IF control statements and display the number of levels of nesting.
3) Program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.
4) Program to evaluate an arithmetic expression involving operators +, -, * and /.
5) Program to recognize the grammar (a^n b, n>=10).

Instructions:

In the examination, a combination of one LEX and one YACC problem has to be asked based on lots. Both will have equal Weightage.

OOP WITH C++ LABORATORY

Subject Code	**: 07MCA37**	I.A. Marks	50

Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

1. Given that an EMPLOYEE class contains the following members:

   <u>Data Members</u>: Employee_Number, Employee_Name, Basic, DA, IT, Net_Sal

   <u>Member Functions</u>: to read data, to calculate Net_Sal and to print data members

   Write a C++ program to read data on N employees and compute the Net_Sal of each employee (DA = 52% of Basic and Income Tax = 30% of the gross salary)
2. Define a STUDENT class with USN, Name, and Marks in 3 tests of a subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of the two better marks for each student. Print the USN, Name and the average marks of all the students.
3. Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a complex number:
   (i) ADD(a, s2) – where ‘a’ is an integer (real part) and s2 is a complex number
   (ii) ADD(s1, s2) – where s1 and s2 are complex numbers
4. Write a C++ program to create a class called LIST (linked list) with member functions to insert an element at the front as well as to delete an element from the front of the list. Demonstrate all the functions after creating a list object.
5. Write a C++ program to create a template function for Quicksort and demonstrate sorting of integers and doubles.
6. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators ‘+’ and ‘-‘:
   (i) s1 = s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on the top of the stack
   (ii) s1 = s1- ; where s1 is an object of the class STACK. ‘-‘operator pops the element.

   Handle the STACK empty and full conditions. Also display the contents of the stack after each operation, by overloading the << operator.
7. Write a C++ program to create a class called DATE. Accept two valid dates in the form dd/mm/yy. Implement the following operations by overloading the operators ‘+’ and ‘-‘. After every operation display the results by overloading the operator <<.
   (i) no_of_days = d1 – d2; where d1 and d2 are DATE objects, and no_of_days is an integer

(ii) d2 = d1 + no_of_days; where d1 is a DATE object and no_of_days is an integer

8. Create a class called MATRIX using two-dimensional array of integers. Implement the following operations by overloading the operator ++ which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the + and – operators respectively. Display the results by overloading the operator <<.
If (m1==m2) then m3 = m1+m2 and m4 = m1-m2 else display error.

9. Write a C++ program to create a class called OCTAL which has the characteristics of an octal number. Implement the following operations by writing an appropriate constructor and an overloaded operator +.
   (i) OCTAL h = x; where x is an integer.
   (ii) int y = h + k; where h is an OCTAL object and k is an integer
   Display the OCTAL result by overloading the operator << . Also display the values of h and y.
10. Write a C++ program to create a class called QUEUE with member functions to add an element and to delete an element from the queue. Using the member functions, implement a queue of integers and double. Demonstrate the operations by displaying the contents of the queue after every operation.
11. Write a C++ program to create a class called DLIST (doubly Linked List) with member functions to insert a node at a specified position and delete a node from a specified position of the list. Demonstrate the operations by displaying the content of the list after every operation.
12. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the semester-wise average age for all UG and PG students separately.
13. Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.
   (i) STRING s1 = “VTU”
   (ii) STRING s2 = “BELGAUM”
   (iii) STRING s3 = s1 + s2 (Use copy constructor)
14. Write a C++ program to create a class called BIN_TREE (Binary Tree) with member functions to perform in-order, preorder and post-order traversals. Create a BIN_TREE object and demonstrate the traversals.
15. Write a C++ program to create a class called EXPRESSION. Using appropriate member functions convert a given valid Infix expression into postfix form. Display the infix and postfix expressions.

DBMS LABORATORY

Subject Code	**: 07MCA38**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

I. Consider the Insurance database given below. The primary keys are underlined and the data types are specified.
PERSON (driver – id #: String, name: string, address: strong)
CAR (Regno: string, model: string, year: int)
ACCIDENT (report-number: int, accd-date: date, location: string)
OWNS (driver-id #:string, Regno:string)
PARTICIPATED (driver-id: string, Regno:string, report-number:int, damage amount:int)
   (i) Create the above tables by properly specifying the primary keys and the foreign keys.
   (ii) Enter at least five tuples for each relation.
   (iii) Demonstrate how you
      a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.
      b. Add a new accident to the database.
   (iv) Find the total number of people who owned cars that were involved in accidents in 2008.
   (v) Find the number of accidents in which cars belonging to a specific model were involved.
   (vi) Generate suitable reports.
   (vii) Create suitable front end for querying and displaying the results.

II. Consider the following relations for an order processing database application in a company.
CUSTOMER (cust #: int , cname: string, city: string)
ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)
ORDER – ITEM (order #: int, item #: int, qty: int)
ITEM (item # : int, unit price: int)
SHIPMENT (order #: int, warehouse#: int, ship-date: date)
WAREHOUSE (warehouse #: int, city: string)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.
(iv) List the order# for orders that were shipped from *all* the warehouses that the company has in a specific city.
(v) Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contain this particular item.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

III. Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)
COURSE (course #:int, cname:string, dept:string)
ENROLL (regno:string, course#:int, sem:int, marks:int)
BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)
TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.
(iv) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
(v) List any department that has *all* its adopted books published by a specific publisher.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

IV. The following tables are maintained by a book dealer.

AUTHOR (author-id:int, name:string, city:string, country:string)
PUBLISHER (publisher-id:int, name:string, city:string, country:string)
CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)
CATEGORY (category-id:int, description:string)
ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
(iv) Find the author of the book which has maximum sales.
(v) Demonstrate how you increase the price of books published by a specific publisher by 10%.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

V. Consider the following database for a banking enterprise

BRANCH(branch-name:string, branch-city:string, assets:real)
ACCOUNT(accno:int, branch-name:string, balance:real)
DEPOSITOR(customer-name:string, accno:int)
CUSTOMER(customer-name:string, customer-street:string, customer-city:string)
LOAN(loan-number:int, branch-name:string, amount:real)
BORROWER(customer-name:string, loan-number:int)

(i) Create the above tables by properly specifying the primary keys and the foreign keys
(ii) Enter at least five tuples for each relation
(iii) Find all the customers who have at least two accounts at the *Main* branch.
(iv) Find all the customers who have an account at *all* the branches located in a specific city.
(v) Demonstrate how you delete all account tuples at every branch located in a specific city.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

Instructions:

1. The exercises are to be solved in an RDBMS environment like Oracle or DB2.
2. Suitable types have to be entered so that queries are executed correctly.
3. Front end may be created using either VB or VAJ or any other similar tool.
4. The student need not create the front end in the examination. The results of the queries may be displayed directly.
5. Relevant queries other than the ones listed along with the exercises may also be asked in the examination.

6. Questions must be asked based on lots.

IV SEMESTER
TOPICS IN ENTERPRISE ARCHITECTURE - 1

Subject Code	**: 07MCA41**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction to Java

Java and Java applications; Java Development Kit (JDK); Java is interpreted, Byte Code, JVM; Object-oriented programming; Simple Java programs.
Data types and other tokens: Boolean variables, int, long, char, operators, arrays, white spaces, literals, assigning values; Creating and destroying objects; Access specifiers.
Operators and Expressions: Arithmetic Operators, Bitwise operators, Relational operators, The Assignment Operator, The ? Operator; Operator Precedence; Logical expression; Type casting; Strings.
Control Statements: Selection statements, iteration statements, Jump Statements.

6 Hours

Classes, Inheritance, Exceptions, Applets

Classes: Classes in Java; Declaring a class; Class name; Super classes; Constructors; Creating instances of class; Inner classes.
Inheritance: Simple, multiple, and multilevel inheritance; Overriding, overloading.
Exception handling**:** Exception handling in Java.
The Applet Class: Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting; Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase(); ApletContext and showDocument(); The AudioClip Interface; The AppletStub Interface; Output to the Console.

6 Hours

Multi Threaded Programming, Event Handling

Multi Threaded Programming: What are threads? How to make the classes threadable; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, read-write problem, producer-consumer problems. Event Handling: Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes.

7 Hours

Swings

Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon; JTextField;The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; JTable.

7 Hours

Java 2 Enterprise Edition Overview, Database Access

Overview of J2EE and J2SE. The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions.

6 Hours

Servlets

Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking.

7 Hours

JSP, RMI

Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects.
Java Remote Method Invocation: Remote Method Invocation concept; Server side, Client side.

6 Hours

Enterprise Java Beans

Enterprise java Beans; Deployment Descriptors; Session Java Bean, Entity Java Bean; Message-Driven Bean; The JAR File.

7 Hours

TEXT BOOKS:

1. Herbert Schildt: **Java The Complete Reference**, 7th Edition, Tata McGraw Hill, 2007.
   (Chapters 1, 2, 3, 4, 5, 6, 8, 10, 11, 21, 22, 29, 30, 31)
2. Jim Keogh: J2EE, **The Complete Reference,** Tata McGraw Hill, 2007.
   (Chapters 5, 6, 11, 12, 15)

REFERENCE BOOKS:

1. Y. Daniel Liang: **Introduction to JAVA Programming**, 6th Edition, Pearson Education, 2007.
2. Stephanie Bodoff et al: **The J2EE Tutorial,** 2nd Edition, Pearson Education, 2004.

SOFTWARE ENGINEERING

Subject Code	**: 07MCA42**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Overview
Introduction: FAQs about software engineering, Professional and ethical responsibility. Socio-Technical systems: Emergent system properties; Systems engineering; Organizations, people and computer systems; Legacy systems.

6 Hours

Critical Systems, Software Processes
Critical Systems: A simple safety-critical system; System dependability; Availability and reliability.
Software Processes: Models, Process iteration, Process activities; The Rational Unified Process; Computer-Aided Software Engineering.

6 Hours

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.

7 Hours

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.

7 Hours

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.

7 Hours

Development

Rapid Software Development: Agile methods; Extreme programming; Rapid application development.
Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution.

6 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.

7 Hours

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.

6 Hours

TEXT BOOKS:

1. Ian Sommerville: **Software Engineering,** 8th Edition, Person Education Ltd., 2007.
   (Chapters-: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 17, 21, 22, 23, 25, 26)

REFERENCE BOOKS:

1. Roger.S.Pressman: **Software Engineering-A Practitioners approach,** 7th Edition,McGraw-Hill, 2007.
2. Shari Lawrence Pfleeger, Joanne M. Atlee : **Software Engineering Theory and Practice**, 3rd Edition, Pearson Education, 2006.
3. Waman S Jawadekar: **Software Engineering Principles and Practice**, Tata McGraw Hill, 2004.

WEB PROGRAMMING

Subject Code	**: 07MCA43**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Fundamentals of Web, XHTML - 1
Internet, WWW, Web Browsers, and Web Servers; URLs; MIME; HTTP; Security; The Web Programmers Toolbox.

XHTML: Origins and evolution of HTML and XHTML; Basic syntax; Standard XHTML document structure; Basic text markup; Images; Hypertext Links; Lists; Tables; Forms; Frames; Syntactic differences between HTML and XHTML.

12 Hours

CSS

Introduction; Levels of style sheets; Style specification formats; Selector forms; Property value forms; Font properties; List properties; Color; Alignment of text; The Box model; Background images; The <span> and <div> tags; Conflict resolution.

6 Hours

Javascript

Overview of Javascript; Object orientation and Javascript; General syntactic characteristics; Primitives, operations, and expressions; Screen output and keyboard input; Control statements; Object creation and modification; Arrays; Functions; Constructor; Pattern matching using regular expressions; Errors in scripts; Examples.

8 Hours

JavaScript and HTML Documents, Dynamic Documents with Javascript

The Javascript execution environment; The Document Object Model; Element access in Javascript; Events and event handling; Handling events from the Body elements, Button elements, Text box and Password elements; The DOM 2 event model; The navigator object; DOM tree traversal and modification.

Introduction to dynamic documents; Positioning elements; Moving elements; Element visibility; Changing colors and fonts; Dynamic content; Stacking elements; Locating the mouse cursor; Reacting to a mouse click; Slow movement of elements; Dragging and dropping elements.

10 Hours

XML

Introduction; Syntax; Document structure; Document Type definitions; Namespaces; XML schemas; Displaying raw XML documents; Displaying XML documents with CSS; XSLT style sheets; XML processors; Web services.

6 Hours

Perl, CGI Programming

Origins and uses of Perl; Scalars and their operations; Assignment statements and simple input and output; Control statements; Fundamentals of arrays; Hashes; References; Functions; Pattern matching; File input and output; Examples. The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module; A survey example; Cookies.

10 Hours

TEXT BOOKS:

1. Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson Education, 2008.
   (Chapters 1 to 9)

REFERENCE BOOKS:

1. M. Deitel, P.J. Deitel, A. B. Goldberg: Internet & World Wide Web How to H program, 3rd Edition, Pearson Education / PHI, 2004.
2. Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2006.
3. Xue Bai et al: The Web Warrior Guide to Web Programming, Thomson, 2003.

DESIGN AND ANALYSIS OF ALGORITHMS

Subject Code	**: 07MCA44**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction

Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental data Structures

6 Hours

Fundamentals of the Analysis of Algorithm Efficiency

Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical analysis of Recursive and Nonrecursive algorithms, Examples

6 Hours

Brute Force

Selection Sort and Bubble Sort, Sequential Search and String Matching, Exhaustive Search

3 Hours

Divide-and-Conquer

Mergesort, Quicksort, Binary Search, Binary tree Traversals and related properties, Multiplication of large integers, Stressen’s Matrix Multiplication

5 Hours

Decrease-and-Conquer

Insertion Sort, Depth First and Breadth First Search, Topological sorting, Algorithms for Generating Combinatorial Objects.

5 Hours

Transform-and-Conquer

Presorting, Balanced Search Trees, Heaps and Heapsort, Problem Reduction

5 Hours

Space and Time Tradeoffs

Sorting by Counting, Input Enhancement in String Matching, Hashing

4 Hours

Dynamic Programming

Computing a binomial coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem and Memory Functions

4 Hours

Greedy Technique

Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees

5 Hours

Limitations of Algorithm Power

Lower-Bound Arguments, Decision Trees, **P**, **NP** and **NP**-Complete Problems

4 Hours

Coping with the Limitations of Algorithm Power

Backtracking, Branch-and-Bound, Approximation Algorithm for NP-Hard problems.

5 Hours

TEXT BOOKS:

1. Anany Levitin: **Introduction to the Design and Analysis of Algorithms**, Pearson Education, 2003.
   (Chapters 1.1-1.4, 2.1-2.5, 3.1, 3.2, 3.4, 4.1-4.5, 5.1-5.4, 6.1, 6.3, 6.4, 6.6, 7.1-7.3, 8.1, 8.2, 8.4, 9.1-9.4, 10.1-10.3, 11.1-11.3)

REFERENCE BOOKS:

1. Coremen T.H., Leiserson C.E., and Rivest R.L.: **Introduction to Algorithms**, PHI, 1998.
2. Horowitz E., Sahani S., Rajasekharan S.: **Computer Algorithms**, Galgotia Publications, 2001.

COMPUTER GRAPHICS AND VISUALIZATION

Subject Code	**: 07MCA451**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction

Applications of computer graphics; A graphics system; Images: Physical and synthetic; The human visual system; The pinhole camera; The synthetic camera model; The programmer's interface; Graphics architectures.
Graphics Programming: The Sierpinski gasket.

7 Hours

The OpenGL

The OpenGL API; Primitives and attributes; Color; Viewing; Control functions; The Gasket program; Polygons and recursion; The three-dimensional gasket.

6 Hours

Input and Interaction

Interaction; Input devices; Clients and servers; Display lists; Programming event-driven input; Menus; Picking; A simple paint program; Animating interactive programs; Design of interactive programs.

6 Hours

Geometric Objects and Transformations

Scalars, points, and vectors; Three-dimensional primitives; Coordinate systems and frames; Modeling a colored cube; Affine transformations; Rotation, translation and scaling; Transformation in homogeneous coordinates; OpenGL transformation matrices; Interfaces to three-dimensional applications.

7 Hours

Viewing

Classical and computer viewing; Positioning of the camera; Simple projections; Projections in OpenGL; Hidden-surface removal; Walking through a scene; Parallel-projection matrices; Perspective-projection matrices; Projections and shadows.

7 Hours

Shading

Light and matter; Light sources; The Phong reflection model; Computation of vectors; Polygonal shading; Approximation of a sphere by recursive subdivisions; Light surfaces in OpenGL; Specification of materials in OpenGL; Shading of the sphere model; Global rendering.

6 Hours

Implementation

The major tasks; Implementation of transformations; Line-segment clipping; Polygon clipping; Clipping of other primitives; Clipping in three dimensions; Hidden-surface removal; Scan conversion; Bresenham's algorithm; Scan conversion of polygons; Antialiasing; Display considerations.

7 Hours

Visualization

Data + Geometry; Height field and contours; Visualizing surfaces and scalar fields; Isosurfaces and marching cubes; Direct volume rendering; Vector-field visualization; Tensor-visualization.

6 Hours

TEXT BOOKS:

1. Edward Angel: **Interactive Computer Graphics** A Top-Down Approach with OpenGL, 2nd Edition, Addison-Wesley, 2000. (Chapters 1, 2, 3, 4, 5, 6, 7, 12)

REFERENCE BOOKS:

1. F.S. Hill,Jr.: **Computer Graphics Using OpenGL,** 2nd Edition, Pearson education / PHI, 2001.
2. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes: **Computer Graphics**, Addison-Wesley, 1997.
3. Donald Hearn and Pauline Baker: **Computer Graphics**- C Version, 2nd Edition, Pearson Education / PHI, 2003.

UNIX SYSTEMS PROGRAMMING

Subject Code	**: 07MCA452**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction:

UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

6 Hours

UNIX Files:

File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links.

6 Hours

UNIX File APIs:

General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files, dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program.

7 Hours

UNIX Processes:

The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes.

7 Hours

Process Control:

Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, waited, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times.

Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp, tcsetpgrp, and tcgetsid Functions, Job Control, Shell Execution of Programs, Orphaned Process Groups.

7 Hours

Signals and Daemon Processes:

Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers.

Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Single-instance daemons; Daemon conventions; Client-Server Model.

7 Hours

Interprocess Communication - 1

Introduction; Pipes, popen, pclose Functions; Coprocesses; FIFOs; XSI IPC; Message Queues; Semaphores.

6 Hours

Network IPC: Sockets

Introduction; Socket Descriptors; Addressing; Connection establishment; Data transfer; Socket options; Out-of-band data; Nonblocking and asynchronous I/O.

6 Hours

TEXT BOOKS:

1 Terrence Chan: **Unix System Programming Using C++,** Prentice-Hall of India / Pearson Education, 1999..
(Chapters 1, 5, 6, 7, 8, 9)

2 W. Richard Stevens, Stephen A. Rago: **Advanced Programming in the UNIX Environment**, 2nd Edition, Pearson Education / Prentice-Hall of India, 2005.
(Chapters 7, 8, 9, 13, 15, 16)

REFERENCE BOOKS:

1. Marc J. Rochkind: **Advanced UNIX Programming**, 2nd Edition, Pearson Education, 2005.
2. Maurice.J. Bach: **The Design of the UNIX Operating System**, Pearson Education / PHI, 1987.
3. Uresh Vahalia: **UNIX Internals**, Pearson Education, 2001.

MULTIMEDIA SYSTEMS

Subject Code	**: 07MCA453**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction, Media and Data Streams, Audio Technology

Multimedia Elements; Multimedia Applications; Multimedia Systems Architecture; Evolving Technologies for Multimedia Systems; Defining Objects for Multimedia Systems; Multimedia Data Interface Standards; The need for Data Compression; Multimedia Databases.

Media: Perception Media, Representation Media, Presentation Media, Storage Media; Characterizing Continuous Media Data Streams.

Sound: Frequency, Amplitude, Sound Perception and Psychoacoustics; Audio Representation on Computers; Three Dimensional Sound Projection; Music and MIDI Standards; Speech Signals; Speech Output; Speech Input; Speech Transmission.

7 Hours

Graphics and Images, Video Technology, Computer-Based Animation

Capturing Graphics and Images Computer Assisted Graphics and Image Processing; Reconstructing Images; Graphics and Image Output Options.

Basics; Television Systems; Digitalization of Video Signals; Digital Television; Basic Concepts; Specification of Animations; Methods of Controlling Animation; Display of Animation; Transmission of Animation; Virtual Reality Modeling Language.

7 Hours

Data Compression - 1

Storage Space; Coding Requirements; Source, Entropy, and Hybrid Coding; Basic Compression Techniques; JPEG: Image Preparation, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-based Mode, Lossless Mode, Hierarchical Mode.

6 Hours

Data Compression - 2

H.261 (Px64) and H.263: Image Preparation, Coding Algorithms, Data Stream, H.263+ and H.263L; MPEG: Video Encoding, Audio Coding, Data Stream, MPEG-2, MPEG-4, MPEG-7; Fractal Compression.

6 Hours

Optical Storage Media

History of Optical Storage; Basic Technology; Video Discs and Other WORMs; Compact Disc Digital Audio; Compact Disc Read Only Memory; CD-ROM Extended Architecture; Further CD-ROM-Based Developments; Compact Disc Recordable; Compact Disc Magneto-Optical; Compact Disc Read/Write; Digital Versatile Disc.

6 Hours

Content Analysis

Simple Vs. Complex Features; Analysis of Individual Images; Analysis of Image Sequences; Audio Analysis; Applications.

6 Hours

Data and File Format Standards

Rich-Text Format; TIFF File Format; Resource Interchange File Format (RIFF); MIDI File Format; JPEG DIB File Format for Still and Motion Images; AVI Indeo File Format; MPEG Standards; TWAIN.

7 Hours

Multimedia Application Design

Multimedia Application Classes; Types of Multimedia Systems; Virtual Reality Design; Components of Multimedia Systems; Organizing Multimedia Databases; Application Workflow Design Issues; Distributed Application Design Issues.

7 Hours

TEXT BOOKS:

1. Ralf Steinmetz, Klara Narstedt: **Multimedia Fundamentals**: Vol 1-Media Coding and Content Processing, 2nd Edition, Pearson Education / PHI, 2003.
   (Chapters 2, 3, 4, 5, 6, 7, 8, 9)
2. Prabhat K. Andleigh, Kiran Thakrar: **Multimedia Systems Design**, PHI, 2003.
   (Chapters 1, 3, 7)

REFERENCE BOOKS:

1. K.R Rao, Zoran S. Bojkovic and Dragorad A. Milovanovic: **Multimedia Communication Systems:** Techniques, Standards, and Networks, Pearson Education, 2002.
2. Nalin K Sharad: **Multimedia information Networking**, PHI, 2002.

PATTERN RECOGNITION

Subject Code	**: 07MCA454**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction
Machine perception, an example; Pattern Recognition System; The Design Cycle; Learning and Adaptation.

6 Hours

Bayesian Decision Theory
Introduction, Bayesian Decision Theory; Continuous Features, Minimum error rate, classification, classifiers, discriminant functions, and decision surfaces; The normal density; Discriminant functions for the normal density.

7 Hours

Maximum-likelihood and Bayesian Parameter Estimation
Introduction; Maximum-likelihood estimation; Bayesian Estimation; Bayesian parameter estimation: Gaussian Case, general theory; Hidden Markov Models.

7 Hours

Non-parametric Techniques
Introduction; Density Estimation; Parzen windows; k_n – Nearest- Neighbor Estimation; The Nearest- Neighbor Rule; Metrics and Nearest-Neighbor Classification.

6 Hours

Linear Discriminant Functions
Introduction; Linear Discriminant Functions and Decision Surfaces; Generalized Linear Discriminant Functions; The Two-Category Linearly Separable case; Minimizing the Perception Criterion Functions; Relaxation Procedures; Non-separable Behavior; Minimum Squared-Error procedures; The Ho-Kashyap procedures.

7 Hours

Stochastic Methods
Introduction; Stochastic Search; Boltzmann Learning; Boltzmann Networks and Graphical Models; Evolutionary Methods.

6 Hours

Non-Metric Methods
Introduction; Decision Trees; CART; Other Tree Methods; Recognition with Strings; Grammatical Methods.

6 Hours

Unsupervised Learning and Clustering
Introduction; Mixture Densities and Identifiability; Maximum-Likelihood Estimates; Application to Normal Mixtures; Unsupervised Bayesian Learning; Data Description and Clustering; Criterion Functions for Clustering.

7 Hours

TEXT BOOKS:

1. Richard O. Duda, Peter E. Hart, and David G.Stork: **Pattern Classification**, 2nd Edition, Wiley-Interscience, 2001.

REFERENCE BOOKS:

1. Earl Gose, Richard Johnsonbaugh, Steve Jost : Pattern Recognition and Image Analysis, Pearson Education / PHI, 2007.

PRINCIPLES OF USER INTERFACE DESIGN

Subject Code	**: 07MCA455**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Human Factors of Interactive Software, Theories, Principles and Guidelines
Introduction, Goals of Systems Engineering, Goals of Interface Design, Motivation for Human Factors in Design, Accommodation of Human Diversity, Goals for the Profession, High Level Theories, Object-Action Interface Model, Principle 1: Recognize the Diversity, Principle 2: Use the Eight Golden Rules of Interface Design, Principle 3: Prevent Errors, Guidelines for Date Display, Guidelines for Data Entry, Balance of Automation and Human Control.

7 Hours

Management Issue
Introduction, Organizational Design to Support Usability, The Three Pillars of Design, Development Methodologies, Ethnographic Observation, Participatory Design, Scenario Development, Social Impact Statement for Early Design Review, Legal Issues.

Expert Reviews, Usability Testing and Laboratories, Surveys, Acceptance Tests, Evaluation During Active Use, Controlled Psychologically Oriented Experiments

6 Hours

Tools and Environments

Introduction, Specification Methods, Interface-Building Tools, Evaluation and Critiquing Tools. Introduction, Examples of Direct-Manipulation Systems, Explanations of Direct Manipulation, Visual Thinking and Icons, Direct Manipulation Programming, Home Automation, Remote Direct Manipulation Virtual Environments.

7 Hours

Menus, Forms, Dialog Boxes and Commands

Task Related Organization, Item Presentation Sequence, Response Time and Display Rate, Fast Movement through Menus, Menu Layout, Form Fillin, Dialog Boxes.

Functionality to support Users Tasks, Command-Organization Strategies, The Benefits of Structure, Naming and Abbreviations, Command Menus, Natural Language in Computing.

6 Hours

Interaction Devices and Response Time

Interaction Devices, Introduction, Keyboards and Function Keys, Pointing Devices, Speech Recognition, Digitization, and Generation, Image and Vide Displays, Printers. Theoretical Foundations, Expectations and Attitudes.

7 Hours

Presentation Styles, Manuals, Help, and Tutorials - 1

Error Messages, Non-anthropomorphic Design, Display Design, Color. Reading from Paper Versus from Displays.

6 Hours

Presentation Styles, Manuals, Help, and Tutorials – 2

Multiple-Window Strategies, Hypermedia and the World Wide Web – 1

Preparation of Printed Manuals, Preparation of Online Facilities.

Multiple-Window Strategies: Introduction, Individual-Window Design, Multiple- window Design, Co-ordination by Tightly Coupled Windows, Image Browsing and Tightly Coupled Windows, Personal Role Management an Elastic Windows.

7 Hours

Multiple-Window Strategies, Hypermedia and the World Wide Web – 2

Genres and Goals and Designers, Users and Their Tasks, Object-Action Interface Model for Web Site Design.

6 Hours

TEXT BOOKS:

1. Ben Shneiderman: **Designing the User Interface**, 3rd Edition, Addison-Wesley, 1998.

(Chapters 1 to 13, and 16)

REFERENCE BOOKS:

1. Alan J Dix et. al.: **Human-Computer Interaction**, II Edition, Prentice Hall, India,1998
2. Eberts: **User Interface Design**, Prentice-Hall, 1994.
3. Wilber O Galitz: **The Essential Guide to User Interface Design** - An Introduction to GUI Design, Principles and Techniques, Wiley-Dreamtech India Pvt. Ltd, 1998.

ADVANCED COMPUTER NETWORKS

Subject Code	**: 07MCA456**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Review of Network Models

Layered tasks; The OSI model and layers in the OSI model; TCP / IP protocol suite; Addressing.

5 Hours

SONET / SDH

Architecture; SONET layers; SONET frames; STS multiplexing; SONET networks; Virtual tributaries.

5 Hours

Frame Relay and ATM

Frame relay; ATM and ATM LANs

4 Hours

IPv6, Address Mapping and Error Reporting

IPv6: Advantages, Packet format, and Extension headers; Transition from IPv4 to IPv6: Dual stack, Tunneling, and Header translation; Address mapping: ARP, RARP, BOOTP, and DHCP; Error reporting: ICMP.

6 Hours

Multicast Routing Protocols

Unicast, multicast and broadcast; Applications; Multicasting routing; Routing protocols.

4 Hours

SCTP

SCTP services; SCTP features; Packet format; An SCTP association; Flow control; Error control; Congestion control.

4 Hours

Congestion Control and Quality of Service

Data traffic; Congestion and congestion control; Congestion control in TCP, Frame relay; Quality of Service; Techniques to improve QoS; Integrated services; Differentiated services.

6 Hours

Multimedia

Digitizing audio and video; Audio and video compression; Streaming stored audio / video; Streaming live audio / video; Real-time interactive audio / video; RTP; RTCP; VoIP.

6 Hours

Mobile Ad-Hoc Networks, Wireless Sensor Networks

Overview of wireless ad-hoc networks; Routing in ad-hoc networks; Routing protocols for ad-hoc networks; Security of ad-hoc networks. Sensor networks and protocol structures; Communication energy model; Clustering protocols; Routing protocols; Zigbee technology and IEEE 802.15.4

12 Hours

TEXT BOOKS:

1. Behrouz A. Forouzan, **Data Communications and Networking**, 4th Edition, Tata McGraw-Hill, 2006.
   (Chapters 2, 17, 18, 20.3, 20.4, 21.1, 21.2, 22.4, 23.4, 24, 29)
2. Nader F. Mir: **Computer and Communication Networks**, Pearson, 2007.
   (Chapters 19, 20 excluding 20.5)

REFERENCES:

1. William Stallings , **Data and Computer Communication**, 5th Edition, Prentice Hall India.
2. William A. Shay, **Understanding Data Communications and Networks**, 2nd Edition, Thomson.
3. Godbole, **Data Communications and Networks**, Tata McGraw-Hill 2002.
4. Micael A. Gallo & William M. Handcock, **Computer Communications and Networking Technologies**, 2003 Edition, Thomson.

JAVA AND J2EE LABORATORY

Subject Code	**: 07MCA46**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

1. a) Write a JAVA Program to demonstrate Constructor Overloading and Method overloading.
   b) Write a JAVA Program to implement Inner class and demonstrate its Access Protections.
2. a) Write a JAVA Program to implement Inheritance.
   b) Write a JAVA Program to implement Exception Handling (Using Nested try catch and finally).
3. a) Write a JAVA Program to create an Interface and implement it in a class.
   b) Write a JAVA Program to create a class (extending Thread) and use methods Thread class to change name, priority, ---- of the current Thread and display the same.
4. a) Write a JAVA Program to create a Scrolling Text using JAVA Applets.
   b)Write a JAVA Program to pass parameters to Applets and display the same.
5. Write a JAVA Program to insert data into Student DATA BASE and retrieve info base on particular queries (Using JDBC Design Front end using Swings).
6. Write a JAVA Program to implement Client Server (Client requests a file,Server responds to client with contents of that file which is then display on the screen by Client – Socket Programming).
7. Write a JAVA Program to implement a simple Client Server Application using RMI.
8. Write a JAVA Servlet Program to implement a dynamic HTML using Servlet (user name and password should be accepted using HTML and displayed using a Servlet).
9. Write a JAVA Servlet Program to Download a file and display it on the screen (A link has to be provided in HTML, when the link is clicked corresponding file has to be displayed on Screen)
10. a) Write a JAVA Servlet Program to implement RequestDispatcher object (use include() and forward() methods).
    b)Write a JAVA Servlet Program to implement and demonstrate get() and Post methods(Using HTTP Servlet Class).
11. Write a JAVA Servlet Program to implement sendRedirect() method(using HTTP Servlet Class).
12. Write a JAVA Servlet Program to implement sessions (Using HTTP Session Interface).
13. a) Write a JAVA JSP Program to print 10 even and 10 odd number.
    b) Write a JAVA JSP Program to implement verification of a particular user login and display a welcome page.

14. Write a JAVA JSP Program to get student information through a HTML and create a JAVA Bean Class, populate Bean and display the same information through another JSP.

WEB PROGRAMMING LABORATORY

Subject Code	**: 07MCA47**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

Note: One exercise must be asked in the examination.
The assignment of the exercise must be based on lots.

1. Develop and demonstrate a XHTML document that illustrates the use external style sheet, ordered list, table, borders, padding, color, and the <span> tag.
2. Develop and demonstrate a XHTML file that includes Javascript script for the following problems:
   a) Input: A number n obtained using prompt
   Output: The first n Fibonacci numbers
   b) Input: A number n obtained using prompt
   Output: A table of numbers from 1 to n and their squares using **alert**
3. Develop and demonstrate a XHTML file that includes Javascript script that uses functions for the following problems:
   a) Parameter: A string
   Output: The position in the string of the left-most vowel
   b) Parameter: A number
   Output: The number with its digits in the reverse order
4. a) Develop and demonstrate, using Javascript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.
   b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)
5. a) Develop and demonstrate, using Javascript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.
   b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom.
6. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
   b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.
7. a) Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.
   b) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.
8. a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.
   b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
9. Write a Perl program to display a digital clock which displays the current time of the server.
10. Write a Perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.
11. Write a PHP program to store current date-time in a COOKIE and display the ‘Last visited on’ date-time on the web page upon reopening of the same page.
12. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.
13. Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.
14. Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

ALGORITHMS LABORATORY

Subject Code	**: 07MCA48**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

Implement the following using C/C++ Language.

1. Implement Recursive Binary search and Linear search and determine the time required to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.

2. Sort a given set of elements using the Heapsort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
3. Sort a given set of elements using Merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.

4. Sort a given set of elements using Selection sort and determine the time required to sort elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.

5. Implement 0/1 Knapsack problem using dynamic programming.

6. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

7. Sort a given set of elements using Quick sort method and determine the time required sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.

8. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

9. a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
   b. Check whether a given graph is connected or not using DFS method.

10. Find a subset of a given set S = {sl,s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions{1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.

11. a. Implement Horspool algorithm for String Matching.
    b. Find the Binomial Co-efficient using Dynamic Programming.

12. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

13. a. Implement Floyd’s algorithm for the All-Pairs- Shortest-Paths problem.
    b. Compute the transitive closure of a given directed graph using Warshall's algorithm.
14. Implement N Queen's problem using Back Tracking.

Note: In the examination questions must be given based on lots.

V SEMESTER

OBJECT-ORIENTED MODELING AND DESIGN PATTERNS

Subject Code	**: 07MCA51**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction, Modeling Concepts, class Modeling
What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history.
Modeling as Design Technique: Modeling; abstraction; The three models.
Class Modeling: Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips.

7 Hours

Advanced Class Modeling, State Modeling
Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips.
State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips.

6 Hours

Advanced State Modeling, Interaction Modeling
Advanced State Modeling: Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models; Practical tips.
Interaction Modeling: Use case models; Sequence models; Activity models.
Use case relationships; Procedural sequence models; Special constructs for activity models.

6 Hours

Process Overview, System Conception, Domain Analysis
Process Overview: Development stages; Development life cycle.
System Conception: Devising a system concept; Elaborating a concept; Preparing a problem statement.
Domain Analysis: Overview of analysis; Domain class model; Domain state model; Domain interaction model; Iterating the analysis.

7 Hours

Application Analysis, System Design
Application Analysis: Application interaction model; Application class model; Application state model; Adding operations.
Overview of system design; Estimating performance; Making a reuse plan; Breaking a system in to sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary conditions; Setting the trade-off priorities; Common architectural styles; Architecture of the ATM system as the example.

7 Hours

Class Design, Implementation Modeling, Legacy Systems
Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example.
Implementation Modeling: Overview of implementation; Fine-tuning classes; Fine-tuning generalizations; Realizing associations; Testing.
Legacy Systems: Reverse engineering; Building the class models; Building the interaction model; Building the state model; Reverse engineering tips; Wrapping; Maintenance.

7 Hours

Design Patterns, Idioms
What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description; Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber; Management Patterns: Command processor; View Handler; Idioms: Introduction; What can idioms provide? Idioms and style; Where to find idioms; Counted Pointer example.

12 Hours

TEXT BOOKS:

1. Michael Blaha, James Rumbaugh: **Object-Oriented Modeling and Design with UML**, 2nd Edition, Pearson Education / PHI, 2005. (Chapters 1 to 17, 23)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: **Pattern-Oriented Software Architecture**, A System of Patterns, Volume 1, John Wiley and Sons, 2006. (Chapters 1, 3.5, 3.6, 4)

REFERENCE BOOKS:

1. Grady Booch et al: **Object-Oriented Analysis and Design with Applications**, 3rd Edition, Pearson, 2007.
2. Mark Priestley: **Practical Object-Oriented Design with UML**, 2nd Edition, Tata McGraw-Hill, 2003.
3. K. Barclay, J. Savage: **Object-Oriented Design with UML and JAVA**, Elsevier, 2008.
4. Booch, G., Rumbaugh, J., and Jacobson, I.: **The Unified Modeling Language** User Guide, 2nd Edition, Pearson, 2005.

5. E. Gamma, R. Helm, R. Johnson, J. Vlissides: **Design Patterns-Elements of Reusable Object-Oriented Software**, Addison-Wesley, 1995.
6. Simon Bennett, Steve McRobb and Ray Farmer: **Object-Oriented Systems Analysis and Design Using UML**, 2nd Edition, Tata McGraw-Hill, 2002.

SYSTEM SIMULATION AND MODELING

Subject Code	**: 07MCA52**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction

When simulation is the appropriate tool and when it 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.

Simulation examples: Simulation of queuing systems; Simulation of inventory systems; other examples of simulation

8 Hours

General Principles, Simulation Software

Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling; List processing. Simulation in Java; Simulation in GPSS.

6 Hours

Statistical Models in Simulation

Review of terminology and concepts; Useful statistical models; Discrete distributions; Continuous distributions; Poisson process; Empirical distributions.

6 Hours

Queuing Models

Characteristics of queuing systems; Queuing notation; Long-run measures of performance of queuing systems; Steady-state behavior of M/G/1 queue; Networks of queues.

6 Hours

Random-Number Generation, Random-Variate Generation

Properties of random numbers; Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers.

Random-Variate Generation: Inverse transform technique; Acceptance-Rejection technique; Special properties.

8 Hours

Input Modeling

Data Collection; Identifying the distribution with data; Parameter estimation; Goodness of Fit Tests; Fitting a non-stationary Poisson process; Selecting input models without data; Multivariate and Time-Series input models**.**

6 Hours

Output Analysis for a Single Model

Types of simulations with respect to output analysis; Stochastic nature of output data; Measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations.

6 Hours

Verification and Validation of Simulation Models, Optimization

Model building, verification and validation; Verification of simulation models; Calibration and validation of models, Optimization via Simulation

6 Hours

TEXT BOOKS:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: **Discrete-Event System Simulation**, 4th Edition, Pearson Education / PHI, 2007.
   (Chapters1, 2, 3, 4.4, 4.5, 5, 6.1 to 6.3, 6.4.1, 6.6, 7, 8, 9, 10, 11, 12.4)

REFERENCE BOOKS:

1. Lawrence M. Leemis, Stephen K. Park: Discrete – **Event Simulation**: A First Course, Pearson / Prentice-Hall, 2006.
2. Averill M. Law: **Simulation Modeling and Analysis**,4th Edition, Tata McGraw-Hill, 2007.

TOPICS IN ENTERPRISE ARCHITECTURE - II

Subject Code	**: 07MCA53**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

The Philosophy of .NET

Understanding the Previous State of Affairs, The .NET Solution, The Building Block of the .NET Platform (CLR,CTS, and CLS), The Role of the

.NET Base Class Libraries, What C# Brings to the Table, An Overview of .NET Binaries (aka Assemblies), the Role of the Common Intermediate Language , The Role of .NET Type Metadata, The Role of the Assembly Manifast, Compiling CIL to Platform –Specific Instructions, Understanding the Common Type System, Intrinsic CTS Data Types, Understanding the Common Languages Specification, Understanding the Common Language Runtime A tour of the .NET Namespaces, Increasing Your Namespace Nomenclature, Deploying the .NET Runtime.

6 Hours

Building C# Applications

The Role of the Command Line Complier (csc.exe), Building C # Application using csc.exe Working with csc.exe Response Files, Generating Bug Reports , Remaining C# Compiler Options, The Command Line Debugger (cordbg.exe) Using the, Visual Studio .NET IDE, Other Key Aspects of the VS.NET IDE, C# "Preprocessor:" Directives, An Interesting Aside: The System. Environment Class

6 Hours

C# Language Fundamentals.

The Anatomy of a Basic C# Class, Creating objects: Constructor Basics, The Composition of a C# Application, Default Assignment and Variable Scope, The C# Member Initialization Syntax, Basic Input and Output with the Console Class, Understanding Value Types and Reference Types, The Master Node: System, Object, The System Data Types (and C# Aliases), Converting Between Value Types and Reference Types: Boxing and Unboxing, Defining Program Constants, C# Iteration Constructs, C# Controls Flow Constructs, The Complete Set of C# Operators, Defining Custom Class Methods, Understating Static Methods, Methods Parameter Modifies, Array Manipulation in C #, String Manipulation in C#, C# Enumerations, Defining Structures in C#, Defining Custom Namespaces.

8 Hours

Object- Oriented Programming with C#

Forms Defining of the C# Class, Definition the "Default Public Interface" of a Type, Recapping the Pillars of OOP, The First Pillars: C#'s Encapsulation Services, Pseudo- Encapsulation: Creating Read-Only Fields, The Second Pillar: C#'s Inheritance Supports, keeping Family Secrets: The " Protected" Keyword, Nested Type Definitions, The Third Pillar: C #'s Polymorphic Support, Casting Between .

6 Hours

Exceptions and Object Lifetime.

Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handing, the System. Exception Base Class, Throwing a Generic Exception, Catching Exception, CLR System – Level Exception (System. System Exception), Custom Application-Level Exception (System. System Exception), Handling Multiple Exception, The Family Block, the Last Chance Exception Dynamically Identifying Application – and System Level Exception Debugging System Exception Using VS. NET, Understanding Object Lifetime, the CIT of "new', The Basics of Garbage Collection,, Finalization a Type, The Finalization Process, Building an Ad Hoc Destruction Method, Garbage Collection Optimizations, The System. GC Type.

6 Hours

Interfaces and Collections

Defining Interfaces Using C# Invoking Interface Members at the object Level, Exercising the Shapes Hierarchy, Understanding Explicit Interface Implementation, Interfaces As Polymorphic Agents, Building Interface Hierarchies, Implementing, Implementation, Interfaces Using VS .NET, understanding the IConvertible Interface, Building a Custom Enumerator (IEnumerable and Enumerator), Building Cloneable objects (ICloneable), Building Comparable Objects (I Comparable), Exploring the system. Collections Namespace, Building a Custom Container (Retrofitting the Cars Type).

6 Hours

Callback Interfaces, Delegates, and Events, Advanced Techniques

Understanding Callback Interfaces, Understanding the .NET Delegate Type, Members of System. Multicast Delegate, The Simplest Possible Delegate Example, , Building More a Elaborate Delegate Example, Understanding Asynchronous Delegates, Understanding (and Using)Events.

The Advances Keywords of C#, A Catalog of C# Keywords Building a Custom Indexer, A Variation of the Cars Indexer Internal Representation of Type Indexer . Using C# Indexer from VB .NET. Overloading operators, The Internal Representation of Overloading Operators, interacting with Overload Operator from Overloaded- Operator- Challenged Languages, Creating Custom Conversion Routines, Defining Implicit Conversion Routines, The Internal Representations of Customs Conversion Routines

8 Hours

Understanding .NET Assembles

Problems with Classic COM Binaries, An Overview of .NET Assembly, Building a Simple File Test Assembly, A C#. Client Application, A Visual Basic .NET Client Application, Cross Language Inheritance, Exploring the CarLibrary's, Manifest, Exploring the CarLibrary's Types, Building the Multifile Assembly ,Using Assembly, Understanding Private Assemblies, Probing for Private Assemblies (The Basics), Private A Assemblies XML Configurations Files, Probing for Private Assemblies (The Details), Understanding Shared Assembly, Understanding Shared Names, Building a Shared Assembly, Understanding Delay Signing, Installing/Removing Shared Assembly, Using a Shared Assembly.

6 Hours

TEXT BOOKS:

1. Andrew Troelsen: **Pro C# with .NET 3.0**, Special Edition, Dream tech Press, India, 2007.
   Chapters: 1 to 11 (up to pp.369)
2. E. Balagurusamy: **Programming in C#,** , 5th Reprint, Tata McGraw Hill, 2004.
   (Programming Examples 3.7, 3.10, 5.5, 6.1, 7.2, 7.4, 7.5, 7.6, 8.1, 8.2, 8.3, 8.5, 87, 8.8, 9.1, 9.2, 9.3, 9.4, 10.2, 10.4, 11.2, 11.4, 12.1, 12.4, 12.5, 12.6, 13.1, 13.2, 13.3, 13.6, 14.1, 14.2, 14.4, 15.2, 15.3, 16.1, 16.2, 16.3, 18.3, 18.5.18.6.)

REFERENCE BOOKS:

1. Tom Archer: **Inside C#**, WP Publishers, 2001.
2. Herbert Schildt: **The Complete Reference C#**, , Tata McGraw Hill, 2004.

INFORMATION RETRIEVAL

Subject Code	**: 07MCA541**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction, Retrieval Strategies - I

Introduction; Retrieval Strategies: Vector Space Model; Probabilistic Retrieval strategies.

7 Hours

Retrieval Strategies – II

Some More Retrieval Strategies: Language Models; Inference Networks; Extended Boolean Retrieval; Latent Semantic Indexing; Neural Networks; Genetic Algorithms; Fuzzy Set Retrieval.

6 Hours

Retrieval Utilities

Relevance feedback; Clustering; Passage-Based Retrieval; N-Grams; Regression Analysis; Thesauri; Semantic Networks; Parsing.

7 Hours

Indexing and Searching

Introduction; Inverted Files; Other indices for text; Boolean queries; Sequential searching; Pattern matching; Structural queries; Compression.

6 Hours

Cross-Language Information Retrieval and Efficiency

Introduction; Crossing the language barrier; Cross-Language retrieval strategies; Cross language utilities. Duplicate Document Detection.

6 Hours

Integrating Structured Data and Text

Review of the relational model; A historical progression; Information retrieval as a relational application; Semi-structured search using a relational schema; Multi-dimensional data model.

6 Hours

Parallel Information Retrieval, Distributed Information Retrieval

Parallel text scanning; Parallel indexing; Clustering and classification; Large parallel systems; A theoretic model of distributed information retrieval; Web search; Result fusion; Peer-to-Peer information systems; Other architectures.

7 Hours

Multimedia IR

Introduction; data modeling; Query languages; Spatial access methods; A general multimedia indexing approach; One-dimensional time series; Two-dimensional color images; Automatic picture extraction.

7 Hours

TEXT BOOKS:

1. David A. Grossman, Ophir Frieder: **Information Retrieval Algorithms and Heuristics**, 2nd Edition, Springer, 2004.
   (Chapters 1, 2, 3, 4, 5, 6, 7, 8)
2. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: **Modern Information Retrieval**, Pearson Education, 1999
   (Chapters 8, 11, 12)

REFERENCE BOOKS:

1. William B. Frakes, Ricardo Baeza-Yates (Editors):**Information Retrieval Data Structures & Algorithm**s, Prentice Hall PTR, 1992.

DATA MINING

Subject Code	**: 07MCA542**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction, Data - I

What is Data Mining? Motivating Challenges; The origins of data mining; Data Mining Tasks. Types of Data; Data Quality.

6 Hours

Data – II

Data Preprocessing; Measures of Similarity and Dissimilarity

6 Hours

Classification

Preliminaries; General approach to solving a classification problem; Decision tree induction; Rule-based classifier; Nearest-neighbor classifier.

8 Hours

Association Analysis - I

Problem Definition; Frequent Itemset generation; Rule Generation; Compact representation of frequent itemsets; Alternative methods for generating frequent itemsets.

6 Hours

Association Analysis - II

FP-Growth algorithm, Evaluation of association patterns; Effect of skewed support distribution; Sequential patterns.

6 Hours

Cluster Analysis

Overview, K-means, Agglomerative hierarchical clustering, DBSCAN, Overview of Cluster Evaluation.

7 Hours

Further Topics in Data Mining

Multidimensional analysis and descriptive mining of complex data objects; Spatial data mining; Multimedia data mining; Text mining; Mining the WWW. Outlier analysis.

7 Hours

Applications

Data mining applications; Data mining system products and research prototypes; Additional themes on Data mining; Social impact of Data mining; Trends in Data mining.

6 Hours

TEXT BOOKS:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson Education, 2007.

(Chapter 1, 2, 4.1 to 4.3, 5.1, 5.2, 6, 8.1 to 8.4, 8.5.1)

2. Jiawei Han and Micheline Kamber: Data Mining – Concepts and Techniques, 2nd Edition, Morgan Kaufmann, 2006.
(Chapters 7.11, 10, 11)

REFERENCE BOOKS:

1. K.P.Soman, Shyam Diwakar, V.Ajay: Insight into Data Mining – Theory and Practice, PHI, 2006.

SUPPLY CHAIN MANAGEMENT

Subject Code	**: 07MCA543**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction to Supply Chain, Performance of Supply Chain

What is a Supply Chain; Decision phases in a supply Chain; Process view of a Supply Chain; The importance of Supply Chain Flows; Examples of Supply Chains.

Competitive and Supply Chain strategies; Achieving strategic fit; Expanding strategic scope.

6 Hours

Supply Chain drivers and Obstacles, Designing Distribution Network

Drivers of Supply Chain Performance; A framework for structuring drivers; Facilities, Inventory, Transportation, and Information; Obstacles to achieve strategic fit.

The role of distribution in the Supply Chain; factors influencing distribution network design; Design options for a distribution network; the value of distributors in the Supply Chain; Distribution Networks in practice.

6 Hours

Network Design

The role of network design in the Supply Chain; Factors influencing Network design Decisions; A framework for Network Design Decisions; Models for facility Location and Capacity Allocation; making Network Design decisions in practice.

The impact of uncertainty on Network design; Discounted cash flow analysis; Representations of uncertainty; Evaluating Network Design decisions using Decision Trees; Making Supply Chain decisions under uncertainty in practice.

7 Hours

Demand Forecasting, Aggregate Planning

The role of forecasting in a Supply Chain; Characteristics of forecast; Components of a forecast and forecasting methods; Basic approach of Demand forecasting; Time series forecasting methods; Measures of forecast errors; The role of aggregate planning in a supply Chain; The aggregate planning problem; Aggregate planning strategies.

7 Hours

Inventory Management

The role of cycle inventory in a supply Chain; Economies of scale to exploit fixed costs, quantity discounts; Short-term discounting; Managing multi-echelon cycle inventory; Estimating cycle inventory related costs in practice.

6 Hours

Transportation

The role of transportation in the Supply Chain; Factors affecting transportation decisions; Modes of transportation and their performance characteristics; Design options for a transportation network; Trade-offs in transportation design; Tailored transportation; Routing and scheduling in transportation; Making transportation decisions in practice.

7 Hours

Pricing and Revenue Management, Coordination

The role of revenue management in Supply Chain; revenue management for multiple customer segments, perishable assets, seasonal demand, and bulk and spot contracts; Using revenue management in practice.

Lack of Supply Chain coordination and Bullwhip effect; Effect of lack of coordination on performance; Obstacles to coordination in the Supply Chain; managerial levers to achieve coordination; Building strategic partnerships and trust within a supply Chain; Achieving coordination in practice.

7 Hours

IT, Internet and Supply Chain

The role of IT in the Supply Chain; The Supply Chain IT framework; CRM; Internal SCM; Supplier Relationship Management; The transaction management foundation; The future if IT in SCM; Supply Chain It in practice.

The role of E-Business in Supply Chain; The E-Business framework; The B2B addition to the E-Business framework; E-Business in practice.

6 Hours

TEXT BOOKS :

1. Sunil Chopra, Pter Meindl: **Supply Chain Management Strategy, Planning, and Operation**, 2nd Edition, Prentice-Hall of India, 2004.
   (Chapters 1, 2, 4, 4, 5, 6, 7, 8.1 to 8.3, 10, 14, 15, 16, 17, 18)

REFERENCE BOOKS:

1. David Simchi-Levi, Philp Kaminky, Edith Simchi-Levi: **Designing and Managing The Supply Chain Concepts, Strategies** & Case Studies, 3rd Edition, tata McGraw Hill, 2003.
2. R.P. Mohanty, S.G. Deshmukh: **Supply Chain Management Theories & Practices**, Bizmantra, 2005.
3. Rahul V. Altekar: **Supply Chain Management Concepts and Cases**, PHI, 2005.
4. M Martin Christopher: **Logistics and Supply Chain Management**, 2nd Edition, Pearson Education, 1998.

NETWORK MANAGEMENT

Subject Code	**: 07MCA544**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction

Analogy of Telephone Network Management, Data and Telecommunication Network Distributed computing Environments, TCP/IP-Based Networks: The Internet and Intranets, Communications Protocols and Standards- Communication Architectures, Protocol Layers and Services; Case Histories of Networking and Management – The Importance of topology , Filtering Does Not Reduce Load on Node, Some Common Network Problems; Challenges of Information Technology Managers, Network Management: Goals, Organization, and Functions- Goal of Network Management, Network Provisioning, Network Operations and the NOC, Network Installation and Maintenance; Network and System Management, Network Management System platform, Current Status and Future of Network Management.

7 Hours

Basic Foundations: Standards, Models, and Language

Network Management Standards, Network Management Model, Organization Model, Information Model – Management Information Trees, Managed Object Perspectives, Communication Model; ASN.1- Terminology, Symbols, and Conventions, Objects and Data Types, Object Names, An Example of ASN.1 from ISO 8824; Encoding Structure; Macros, Functional Model.

6 Hours

SNMPv1 Network Management - 1

Managed Network: The History of SNMP Management, Internet Organizations and standards, Internet Documents, The SNMP Model, The Organization Model, System Overview.

6 Hours

SNMPv1 Network Management – 2

The Information Model – Introduction, The Structure of Management Information, Managed Objects, Management Information Base.

The SNMP Communication Model – The SNMP Architecture, Administrative Model, SNMP Specifications, SNMP Operations, SNMP MIB Group, Functional Model.

7 Hours

SNMP Management – RMON

Remote Monitoring, RMON SMI and MIB, RMONI1- RMON1 Textual Conventions, RMON1 Groups and Functions, Relationship Between Control and Data Tables, RMON1 Common and Ethernet Groups, RMON Token Ring Extension Groups, RMON2 – The RMON2 Management Information Base, RMON2 Conformance Specifications; ATM Remote Monitoring, A Case Study of Internet Traffic Using RMON.

6 Hours

Broadband Network Management: ATM Networks

Broadband Networks and Services, ATM Technology – Virtual Path-Virtual Circuit, TM Packet Size, Integrated Service, SONET, ATM LAN Emulation, Virtual LAN; ATM Network Management – The ATM Network Reference Model, The Integrated Local Management Interface, The ATM Management Information Base, The Role of SNMP and ILMI in ATM Management, M1 Interface: Management of ATM Network Element, M2 Interface: Management of Private Networks, M3 Interface: Customer Network Management of Public Networks, M4 Interface: Public Network Management, Management of LAN Emulation, ATM Digital Exchange Interface Management.

6 Hours

Broadband Network Management

Broadband Access Networks and Technologies – Broadband Access Networks, roadband Access Technology; HFCT Technology – The Broadband LAN, The Cable Modem, The Cable Modem Termination System, The HFC Plant, The RF Spectrum for Cable Modem; Data Over Cable Reference Architecture; HFC Management – Cable Modem and CMTS Management, HFC Link Management, RF Spectrum Management, DSL Technology; Asymmetric Digital Subscriber Line Technology – Role of the ADSL Access Network in an Overall Network, ADSL Architecture, ADSL Channeling Schemes, ADSL Encoding Schemes; ADSL Management – ADSL Network Management Elements, ADSL Configuration Management, ADSL Fault Management, ADSL Performance Management, SNMP-Based ADSL Line MIB, MIB Integration with Interfaces Groups in MIB-2, ADSL Configuration Profiles.

6 Hours

Network Management Applications

Configuration Management- Network Provisioning, Inventory Management, Network Topology, Fault Management- Fault Detection, Fault Location and Isolation Techniques, Performance Management – Performance Metrics, Data Monitoring, Problem Isolation, Performance Statistics; Event Correlation Techniques – Rule-Based Reasoning, Model-Based Reasoning, Case-Based Reasoning, Codebook correlation Model, State Transition Graph Model, Finite State Machine Model, Security Management – Policies and Procedures, Security Breaches and the Resources Needed to Prevent Them, Firewalls, Cryptography, Authentication and Authorization, Client/Server Authentication Systems, Messages Transfer Security, Protection of Networks from Virus Attacks, Accounting Management, Report Management, Policy-Based Management, Service Level Management.

8 Hours

TEXT BOOKS:

1. Mani Subramanian: **Network Management-** Principles and Practice, Pearson Education Publication, 2003.

REFERENCE BOOKS:

1. J. Richard Burke: Network management Concepts and Practices: a Hands-On Approach, PHI, 2008.

COMPILER DESIGN

Subject Code	**: 07MCA545**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction, Lexical analysis

Language processors; The structure of a Compilers; The evolution of programming languages; The science of building a compiler; Applications of Compiler technology; Programming language basics;

Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.

8 Hours

Syntax Analysis – I

Introduction; Context-free Grammars; Writing a Grammar; Top-down Parsing.

6 Hours

Syntax Analysis – II

Bottom-up Parsing; Introduction to LR Parsing: Simple LR.

6 Hours

Syntax Analysis – III

More powerful LR parsers; Using ambiguous grammars; Parser Generators.

6 Hours

Syntax-Directed Translation

Syntax-Directed definitions; Evaluation order for SDDs; Applications of Syntax-directed translation; Syntax-directed translation schemes.

6 Hours

Intermediate Code Generation

Variants of syntax trees; Three-address code; Types and declarations; Translation of expressions; Type checking; Control flow; Back patching; Switch statements; Intermediate code for procedures.

8 Hours

Run-Time Environments

Storage Organization; Stack allocation of space; Access to non-local data on the stack; Heap management; Introduction to garbage collection.

6 Hours

Code Generation

Issues in the design of Code Generator; The Target language; Addresses in the target code; Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator.

6 Hours

TEXT BOOKS:

1. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman: **Compilers- Principles, Techniques and Tools**, 2nd Edition, Addison-Wesley, 2007.
   (Chapters 1, 3.1 to 3.4, 4, 5.1 to 5.4, 6, 7.1 to 7.5, 8.1 to 8.6)

REFERENCE BOOKS:

1. Charles N. Fischer, Richard J. leBlanc, Jr.: **Crafting a Compiler with C**, Pearson Education, 1991.
2. Andrew W Apple: **Modern Compiler Implementation in C**, Cambridge University Press, 1997.
3. Kenneth C Louden: **Compiler Construction Principles & Practice**, Thomson Education, 1997.

SOFTWARE ARCHITECTURES

Subject Code	**: 07MCA546**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction

The Architecture Business Cycle: Where do architectures come from? Software processes and the architecture business cycle; What makes a "good" architecture? What software architecture is and what it is not; Other points of view; Architectural patterns, reference models and reference architectures; Importance of software architecture; Architectural structures and views.

6 Hours

Architectural Styles and Case Studies

Architectural styles; Pipes and filters; Data abstraction and object-oriented organization; Event-based, implicit invocation; Layered systems; Repositories; Interpreters; Process control; Other familiar architectures; Heterogeneous architectures.

Case Studies: Keyword in Context; Instrumentation software; Mobile robotics; Cruise control; Three vignettes in mixed style.

7 Hours

Quality

Functionality and architecture; Architecture and quality attributes; System quality attributes; Quality attribute scenarios in practice; Other system quality attributes; Business qualities; Architecture qualities.

Achieving Quality: Introducing tactics; Availability tactics; Modifiability tactics; Performance tactics; Security tactics; Testability tactics; Usability tactics; Relationship of tactics to architectural patterns; Architectural patterns and styles.

6 Hours

Architectural Patterns

Introduction; From mud to structure: Layers, Pipes and Filters, Blackboard.
Distributed Systems: Broker; Interactive Systems: MVC, Presentation-Abstraction-Control. Adaptable Systems: Microkernel; Reflection.

20 Hours

Some Design Patterns

Structural decomposition: Whole – Part; Organization of work: Master – Slave; Access Control: Proxy.

6 Hours

Designing and Documenting Software Architecture

Architecture in the life cycle; Designing the architecture; forming the team structure; creating a skeletal system. Uses of architectural documentation;

Views; Choosing the relevant views; Documenting a view; Documentation across views.

7 Hours

TEXT BOOKS:

1. Len Bass, Paul Clements, Rick Kazman: **Software Architecture in Practice**, 2nd Edition, Pearson Education, 2003.
(Chapters 1, 2, 4, 5, 7, 9)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: **Pattern-Oriented Software Architecture, A System of Patterns**, Volume 1, John Wiley and Sons, 2006.
(Chapters 2, 3.1 to 3.4)
3. Mary Shaw and David Garlan: **Software Architecture-Perspectives on an Emerging Discipline**, Prentice-Hall of India / Pearson Education, 2007.
(Chapters 1.1, 2, 3)

REFERENCE BOOKS:

1. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

Web site for Patterns: http://www.hillside.net/patterns/

INFORMATION AND NETWORK SECURITY

Subject Code	**: 07MCA551**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Planning for Security
Introduction; Information Security Policy, Standards, and Practices; The Information Security Blue Print; Contingency plan and a model for contingency plan.

6 Hours

Security Technology
Introduction; Physical design; Firewalls; Protecting Remote Connections.
Introduction; Intrusion Detection Systems (IDS); Honey Pots, Honey Nets, and Padded cell systems; Scanning and Analysis Tools.

12 Hours

Cryptography
Introduction; A short History of Cryptography; Principles of Cryptography; Cryptography Tools; Attacks on Cryptosystems.

8 Hours

Introduction to Network Security, Authentication Applications
Attacks , services, and Mechanisms; Security Attacks; Security Services; A model for Internetwork Security; Internet Standards and RFCs.
Kerberos, X.509 Directory Authentication Service.

8 Hours

Electronic Mail Security
Pretty Good Privacy (PGP); S/MIME.

6 Hours

IP Security
IP Security Overview; IP Security Architecture; Authentication Header; Encapsulating Security Payload; Combining Security Associations; Key Management.

6 Hours

Web Security
Web security requirements; Secure Socket layer (SSL) and Transport layer Security (TLS); Secure Electronic Transaction (SET).

6 Hours

TEXT BOOKS:

1. Michael E. Whitman and Herbert J. Mattord: **Principles of Information Security**, 2nd Edition, Thomson, 2005.
(Chapters 5, 6, 7, 8; Exclude the topics not mentioned in the syllabus)
2. William Stallings: Network Security Essentials: **Applications and Standards**, Pearson Education, 2000.
(Chapters: 1, 4, 5, 6, 7, 8)

REFERENCE BOOK:

1. Behrouz A. Forouzan: **Cryptography and Network Security**, Special Indian Edition, Tata McGraw-Hill, 2007.

SOFTWARE TESTING

Subject Code	**: 07MCA552**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Basics of Software Testing

Human Errors and Testing; Software Quality; Requirements, Behavior and Correctness; Correctness versus Reliability; Testing and Debugging; Test Metrics; Software and Hardware Testing; Testing and Verification; Defect Management; Execution History; Test-generation Strategies, Static Testing. Model-Based Testing and Model Checking; Control-Flow Graph; Types of Testing; The Saturation Effect.

12 Hours

Test Generation from Requirements, Structural Testing

Introduction; The Test-Selection Problem; Equivalence Partitioning; Boundary Value Analysis; Category-Partition Method. Cause-Effect Graphing, Test Generation from Predicates, Structural Testing: Overview; Statement testing; Branch testing; Condition testing, Path testing; Procedure call testing; Comparing structural testing criteria; The infeasibility problem.

20 Hours

Dependence, Data Flow Models, and Data Flow Testing

Definition-Use pairs; Data flow analysis; Classic analyses; From execution to conservative flow analysis; Data flow analysis with arrays and pointers; Inter-procedural analysis; Overview of data flow testing; Definition-Use associations; Data flow testing criteria; Data flow coverage with complex structures; The infeasibility problem.

5 Hours

Test Case Selection and Adequacy, Test Execution

Overview; Test specification and cases; Adequacy criteria; Comparing criteria; Overview of test execution; From test case specification to test cases; Scaffolding; Generic versus specific scaffolding; Test oracles; Self-checks as oracles; Capture and replay.

5 Hours

Process

Test and analysis activities within a software process: The quality process; Planning and monitoring; Quality goals; Dependability properties; Analysis; Testing; Improving the process; Organizational factors.

Integration and component-based software testing: Overview; Integration testing strategies; Testing components and assemblies.

System, Acceptance and Regression Testing: Overview; System testing; Acceptance testing; Usability; Regression testing; Regression test selection techniques; Test case prioritization and selective execution.

10 Hours

TEXT BOOKS:

1. Aditya P Mathur: **Foundations of Software Testing**, Pearson Education, 2008. (Chapters 1 excluding 1.15, 1.16, 1.17, 2, 6)
2. Mauro Pezze, Michal Young: **Software Testing and Analysis – Process, Principles and Techniques**, John Wiley & Sons, 2008. (Chapters 4, 6, 9, 12, 13, 17, 21, 22)

REFERENCE BOOKS:

1. Srinivasan Desikan, Gopalaswamy Ramesh: **Software testing Principles and Practices**, 2nd Edition, Pearson, 2007.
2. Ron Patton: **Software Testing**, 2nd edition, Pearson, 2004.
3. Brian Marrick:**The Craft of Software Testing**, Pearson, 1995.

SERVICE ORIENTED ARCHITECTURE

Subject Code	**: 07MCA553**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction o SOA, Evolution of SOA

Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits of SOA; An SOA timeline (from XML to Web services to SOA); The continuing evolution of SOA (Standards organizations and Contributing vendors); The roots of SOA (comparing SOA to Past architectures).

8 Hours

Web Services and Primitive SOA

The Web services framework; Services (as Web services); Service descriptions (with WSDL); Messaging (with SOAP).

6 Hours

Web Services and Contemporary SOA

Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography.

Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing

12 Hours

Principles of Service – Orientation

Services-orientation and the enterprise; Anatomy of a service-oriented architecture; Common Principles of Service-orientation; How service orientation principles inter-relate; Service-orientation and object-orientation; Native Web service support for service-orientation principles.

7 Hours

Service Layers

Service-orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios.

6 Hours

Business Process Design

WS-BPEL language basics; WS-Coordination overview; Service-oriented business process design; WS-addresing language basics; WS-ReliableMessaging language basics

7 Hours

SOA Platforms

SOA platform basics; SOA support in J2EE; SOA support in .NET; Integration considerations.

6 Hours

TEXT BOOKS:

1. Thomas Erl: Service-Oriented Architecture – **Concepts, Technology, and Design,** Pearson Education, 2005.

REFERENCE BOOKS:

1. Eric Newcomer, Greg Lomow: **Understanding SOA with Web Services**, Pearson Education, 2005.

MOBILE COMPUTING

Subject Code	**: 07MCA554**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Mobile Devices and Systems, Architectures

Mobile phones, Digital Music Players, Handheld Pocket Computers, Handheld Devices, Operating Systems, Smart Systems, Limitations of Mobile Devices, Automotive Systems GSM – Services and System

Architectures, Radio Interfaces, Protocols, Localization, Calling, Handover, General Packet Radio Service.

8 Hours

Wireless Medium Access Control and CDMA – based Communication

Medium Access Control, Introduction to CDMA – based Systems, OFDM

6 Hours

Mobile IP Network Layer, Mobile Transport Layer

IP and Mobile IP Network Layers Packet Delivery and Handover Management, Registration, Tunneling and Encapsulation, Route Optimization, Dynamic Host Configuration Protocol.

Indirect TCP, Snooping TCP, Mobile TCP, Other Methods of TCP – layer Transmission for Mobile Networks.

7 Hours

Databases

Database Hoarding Techniques, Data Caching, Client – Server Computing and Adaptation, Transactional Models, Query Processing, Data Recovery Process, Issues relating to Quality of Service.

5 Hours

Data Dissemination and Broadcasting Systems

Communication Asymmetry, Classification of Data – Delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing Techniques, Digital Audio Broadcasting, Digital video Broadcasting.

5 Hours

Data Synchronization in Mobile Computing Systems

Synchronization, Synchronization Protocols, SyncML – Synchronization Language for Mobile Computing, Synchronized Multimedia Markup Language (SMIL).

6 Hours

Mobile Devices, Server and Management, Wireless LAN, Mobile Internet Connectivity and Personal Area Network

Mobile agent, Application Server, Gateways, Portals, Service Discovery, Device Management, Mobile File Systems.

Wireless LAN (WiFi) Architecture and Protocol Layers, WAP 1.1 and WAP 2.0 Architectures, Bluetooth – enabled Devices Network, Zigbee.

8 Hours

Mobile Application languages – XML, Java, J2ME and JavaCard, Mobile Operating Systems

Introduction, XML, JAVA, Java 2 Micro Edition (J2ME), JavaCard
Operating System, PalmOS, Windows CE, Symbian OS, Linux for Mobile Devices.

7 Hours

TEXT BOOK:

1. Raj Kamal: **Mobile Computing**, Oxford University Press, 2007.

REFERENCE BOOKS:

1. Asoke Talkukder, Roopa R Yavagal: **Mobile Computing – Technology**, Applications and Service Creation, Tata McGraw Hill, 2005.
2. Reza B'Far: Mobile Computing Principles – **Designing and Developing Mobile Applications with UML and XML**, 5th Edition, Cambridge University press, 2006.
3. Uwe Hansmann, Lothat Merk, Martin S Nicklous and Thomas Stober: **Principles of Mobile Computing,** 2nd Edition, Springer International Edition, 2003.
4. Schiller: **Mobile Communication**, Pearson Publication, 2004.

ARTIFICIAL INTELLIGENCE

Subject Code	**: 07MCA555**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction
What is AI? Intelligent Agents: Agents and environment; Rationality; the nature of environment; the structure of agents. Problem-solving: Problem-solving agents; Example problems; Searching for solution; uninformed search strategies.

7 Hours

Informed Search, Exploration, Constraint Satisfaction, Adversial Search
Informed search strategies; Heuristic functions; On-line search agents and unknown environment. Constraint satisfaction problems; Backtracking search for CSPs. Adversial search: Games; Optimal decisions in games; Alpha-Beta pruning.

7 Hours

Logical Agents
Knowledge-based agents; The wumpus world as an example world; Logic; propositional logic Reasoning patterns in propositional logic; Effective propositional inference; Agents based on propositional logic.

6 Hours

First-Order Logic, Inference in First-Order Logic
Representation revisited; Syntax and semantics of first-order logic; Using first-order logic; Knowledge engineering in first-order logic. Propositional versus first-order inference; Unification and lifting; Forward chaining; Backward chaining; Resolution.

12 Hours

Knowledge Representation
Ontological engineering; Categories and objects; Actions, situations, and events; Mental events and mental objects; The Internet shopping world; Reasoning systems for categories; Reasoning with default information; Truth maintenance systems.

7 Hours

Planning, Uncertainty, Probabilistic Reasoning
Planning: The problem; Planning with state-space approach; Planning graphs; Planning with propositional logic.
Uncertainty: Acting under certainty; Inference using full joint distributions; Independence; Bayes' rule and its use.
Probabilistic Reasoning: Representing knowledge in an uncertain domain; The semantics of Bayesian networks; Efficient representation of conditional distributions; Exact inference in Bayesian networks.

7 Hours

Learning, AI: Present and Future
Learning: Forms of Learning; Inductive learning; Learning decision trees; Ensemble learning; Computational learning theory.
AI: Present and Future: Agent components; Agent architectures; Are we going in the right direction? What if AI does succeed?

6 Hours

TEXT BOOKS:

1. Stuart Russel, Petr Norvig: **Artificial Intelligence A Modern Approach,** 2nd Edition, Pearson Education, 2003.
   (Chapters 1.1, 2, 3.1 to 3.4, 4.1, 4.2, 4.5, 5.1, 5.2, 6.1, 6.2, 6.3, 7, 8, 9, 10, 11.1, 11.2, 11.4, 11.5, 13.1, 13.4, 13.5, 13.6, 14.1, 14.2, 14.3, 14.4, 18, 27)

REFERENCE BOOKS:

1. Elaine Rich, Kevin Knight: **Artificial Intelligence,** 2nd Edition, Tata McGraw Hill, 1991.
2. Nils J. Nilsson: **Principles of Artificial Intelligence**, Elsevier, 1980.

WEB 2.0 AND RICH INTERNET APPLICATIONS

Subject Code	**: 07MCA556**	I.A. Marks	50
Hours/Week	: 04	Exam Hours	03
Total Hours	: 52	Exam Marks	100

Introduction, Web Services

What is Web 2.0?, Folksonomies and Web 2.0, Software As a Service (SaaS), Data and Web 2.0, Convergence, Iterative development, Rich User experience, Multiple Delivery Channels, Social Networking.

Web Services: SOAP, RPC Style SOAP, Document style SOAP, WSDL, REST services, JSON format, What is JSON?, Array literals, Object literals, Mixing literals, JSON Syntax, JSON Encoding and Decoding, JSON versus XML.

8 Hours

Building Rich Internet Applications with AJAX

Building Rich Internet Applications with AJAX: Limitations of Classic Web application model, AJAX principles, Technologies behind AJAX, Examples of usage of AJAX, Dynamic web applications through Hidden frames for both GET and POST methods.

IFrames, Asynchronous communication and AJAX application model, XMLHTTP Object – properties and methods, handling different browser implementations of XMLHTTP, The same origin policy, Cache control, AJAX Patterns (Only algorithms – examples not required): Predictive fetch pattern, Submission throttling pattern, Periodic refresh, Multi stage download, Fall back patterns.

12 Hours

Building Rich Internet Applications with Flex

Flash player, Flex framework, MXML and Actionscript, Working with Data services, Understanding differences between HTML and Flex applications, Understanding how Flex applications work, Understanding Flex and Flash authoring, MXML language, a simple example.

Using Actionscript, MXML and Actionscript correlations. Understanding Actionscript 3.0 language syntax: Language overview, Objects and Classes, Packages and namespaces, Variables & scope of variables, case sensitivity and general syntax rules, Operators, Conditional, Looping, Functions, Nested functions, Functions as Objects, Function scope, OO Programming in Actionscript: Classes, Interfaces, Inheritance, Working with String objects, Working with Arrays, Error handling in Actionscript: Try/Catch, Working with XML

Framework fundamentals, Understanding application life cycle, Differentiating between Flash player and Framework, Bootstrapping Flex applications, Loading one flex application in to another, Understanding application domains, Understanding the preloader.

Managing layout, Flex layout overview, Working with children, Container types, Layout rules, Padding, Borders and gaps, Nesting containers, Making fluid interfaces.

Working with UI components: Understanding UI Components, Creating component instances, Common UI Component properties, Handling events, Button, Value selectors, Text components, List based controls, Data models and Model View Controller, Creating collection objects, Setting the data provider, Using Data grids, Using Tree controls, Working with selected values and items, Pop up controls, Navigators, Control bars

Working with data: Using data models, Using XML, Using Actionscript classes, Data Binding.

24 Hours

Building Advanced Web 2.0 applications

Definition of mash up applications, Mash up Techniques, Building a simple mash up application with AJAX, Remote data communication, strategies for data communication, Simple HTTPServices, URLLoader in Flex, Web Services in Flex, Examples: Building an RSS reader with AJAX, Building an RSS reader with Flex.

8 Hours

TEXT BOOKS:

1. Nicholas C Zakas et al: Professional AJAX, Wrox publications, 2006. (Chapters 1 to 4, Chapter 6 pp157-166, Chapter 7 pp191-196)
2. Chafic Kazoun: Programming Flex 2, O'Reilly publications, 2007. (Chapters 1, Chapters 3 to 7, Chapter 12, Chapter 16 pp380-403)
   Francis Shanahan: Mashups, Wrox, 2007.
   (Chapters 1, 6)

REFERENCE BOOKS:

1. Thomas A. Powel: Ajax **The Complete reference**, McGraw Hill, 2008.
2. Gottfried Vossen, Stephan Hagemann: **Unleashing Web 2.0 From Concepts to Creativity**, Elsevier, 2007.
3. Colin Moock: Essential Actionscript 3.0, O'Reilly Publications, 2007.
4. Steven Holzner : **Ajax Bible** Wiley India , 2007.
5. Justin Gehtland et al: **A Web 2.0 primer Pragmatic Ajax**, SPD Publications, 2006.
6. Eric Van derVlist et al: **Professional Web 2.0 Programming**, Wiley India, 2007.

SOFTWARE DESIGN LABORATORY

Subject Code	**: 07MCA56**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

The student has to draw the necessary UML diagrams using any suitable UML Drawing Tool and implement in Java OR C++ OR C# a program to demonstrate the Design Pattern specified by the Examiner.

The Design Pattern is allotted based on lots from the following list:

1) Expert
2) Controller
3) Publisher-Subscriber
4) Command
5) Forward-Receive
6) Client-Dispatcher
7) Proxy
8) Façade
9) Polymorphism

. NET LABORATORY

Subject Code	**: 07MCA57**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

1. Write a Program in C# to Check whether a number is Palindrome or not.
2. Write a Program in C# to demonstrate Command line arguments Processing.
3. Write a Program in C# to find the roots of Quadratic Equation.
4. Write a Program in C# to demonstrate boxing and unBoxing.
5. Write a Program in C# to implement Stack operations.
6. Write a program to demonstrate Operator overloading.
7. Write a Program in C# to find the second largest element in a single dimensional array.
8. Write a Program in C# to multiply to matrices using Rectangular arrays.
9. Find the sum of all the elements present in a jagged array of 3 inner arrays.
10. Write a program to reverse a given string using C#.

11. Using Try, Catch and Finally blocks write a program in C# to demonstrate error handling.
12. Design a simple calculator using Switch Statement in C#.
13. Demonstrate Use of Virtual and override key words in C# with a simple program
14. Implement linked lists in C# using the existing collections name space.
15. Write a program to demonstrate abstract class and abstract methods in C#.
16. Write a program in C# to build a class which implements an interface which is already existing.
17. Write a program to illustrate the use of different properties in C#.
18. Demonstrate arrays of interface types with a C# program.

MINI PROJECT

Subject Code	**: 07MCA58**	I.A. Marks	50
Hours/Week	: 03	Exam Hours	03
Total Hours	: 42	Exam Marks	50

NOTES:

- A team of **TWO** students must develop the mini project. However, during the examination, each student must demonstrate the project individually.
- The team may implement a mini project of their choice.
- The team must submit a **Brief Project Report** (25 to 30 Pages) that must include the following:
  - Introduction
  - Requirements
  - Software Development Process Model Adopted
  - Analysis and Design Models
  - Implementation
  - Testing
- The Report must be valuated for 10 marks, Demonstration for 30 marks and Viva for 10 marks.