# SCHEME OF STUDIES
## MCA Degree Programme
### 1st Year
#### SEMESTER – I
<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-101</td>
<td>Computer Programming</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>MCA-102</td>
<td>Discrete Structures</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>MCA-103</td>
<td>Relational DBMS</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>EN-1101</td>
<td>Technical Communication</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>MA-202</td>
<td>Applied Numerical Methods</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>MCA-104</td>
<td>Data Communication and Networking</td>
<td>3-0-0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-151</td>
<td>Computer Programming Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>MCA-153</td>
<td>Relational DBMS Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>MA-252</td>
<td>Applied Numerical Methods Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total (PRACTICAL/DRAWING/DESIGN)</strong></td>
<td></td>
<td><strong>28</strong></td>
</tr>
</tbody>
</table>

#### SEMESTER – II
<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-105</td>
<td>Object Oriented Programming USING C++</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>MCA-106</td>
<td>Web Development</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>BA-247</td>
<td>Accounting and Financial Management</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>MCA-107</td>
<td>Computer Organization and Architecture</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>MCA-108</td>
<td>Data Structures and its application</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>EC-307</td>
<td>Wireless Communication</td>
<td>3-1-0</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-156</td>
<td>Web Development Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>MCA-155</td>
<td>Object Oriented Programming USING C++ Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>MCA-158</td>
<td>Data Structures and Its Applications Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total (PRACTICAL/DRAWING/DESIGN)</strong></td>
<td></td>
<td><strong>27</strong></td>
</tr>
</tbody>
</table>
### SCHEME OF STUDIES
MCA Degree Programme

#### 2nd Year (1st yr for Lateral entry)

<table>
<thead>
<tr>
<th>Semester – III (semester – I for Lateral Entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.N.</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

#### PRACTICAL/DRAWING/DESIGN

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-251</td>
<td>Computer Graphics Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>MCA-254</td>
<td>Linux and Shell Programming Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>MCA-256</td>
<td>Core and Advance Java Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>18-5-8(31)</td>
<td>25</td>
</tr>
</tbody>
</table>

#### Semester – IV (Semester- II for Lateral Entry)

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-207</td>
<td>Rapid Application Development</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>MCA-208</td>
<td>Programming Using C#</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>MCA-209</td>
<td>Data Mining and Data Warehousing</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>MCA-210</td>
<td>Software Project Management</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>MCA-211</td>
<td>IT Management</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>MCA-205</td>
<td>Multimedia Systems</td>
<td>3-0-0</td>
<td>3</td>
</tr>
</tbody>
</table>

#### PRACTICAL/DRAWING/DESIGN

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-257</td>
<td>Rapid Application Development Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>MCA-258</td>
<td>C# Programming Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>MCA-255</td>
<td>multimedia System Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>18-8-6(32)</td>
<td>29</td>
</tr>
</tbody>
</table>
### SCHEME OF STUDIES

**MCA Degree Programme**

**3rd Yr (2nd Yr for Lateral entry)**

#### SEMESTER – V (Semester III for lateral Entry)

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-302</td>
<td>System Network Administration</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Elective -1</td>
<td>Artificial Intelligence Lab</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Elective -2</td>
<td>Elective-3</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>MCA-303</td>
<td>Introduction to ERP</td>
<td>3-1-0</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>MCA-301</td>
<td>Artificial Intelligence</td>
<td>3-1-0</td>
<td>4</td>
</tr>
</tbody>
</table>

#### PRACTICAL/DRAWING/DESIGN

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-352</td>
<td>System Network Administration Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MCA-351</td>
<td>Artificial Intelligence Lab</td>
<td>0-1-2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MCA-381</td>
<td>Minor project</td>
<td>0-0-4</td>
<td>2</td>
</tr>
</tbody>
</table>

**18-8-6(32)**

**SEMESTER – VI ( Semester IV for lateral entry)**

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCA-371</td>
<td>Internship</td>
<td>0-0-32</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>MCA-391</td>
<td>Seminar***</td>
<td>0-0-4</td>
<td>2</td>
</tr>
</tbody>
</table>

**0-0-32(32)**
## LIST OF ELECTIVES
### MCA Degree Programme

### Elective -1

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CA-1325</td>
<td>Cryptography and Data Compression</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>CA-1408</td>
<td>Object Oriented Software Engineering &amp; UML</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CA-1309</td>
<td>Network Security &amp; Management</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>CA-1425</td>
<td>Information Storage and Management</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>CA-1401</td>
<td>Introduction to XML</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>CA-1323</td>
<td>Advanced Computer Architecture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ELECTIVE -2

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CA-1328</td>
<td>Digital Image Processing</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>CA-1407</td>
<td>MOBILE COMPUTING</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CA-1406</td>
<td>EMBEDDED SYSTEM DESIGN</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>CA-1307</td>
<td>Neural Network</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>CA-1402</td>
<td>Software Testing</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>CA-1324</td>
<td>Advanced Database Management System</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Elective -3

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Course No.</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Cr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CA-1427</td>
<td>Android Application development</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>CA-1326</td>
<td>Expert System</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CA-1327</td>
<td>Natural language processing</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>CA-1422</td>
<td>Soft Computing</td>
<td>3-0-0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>CA-1421</td>
<td>Compiler Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CA-1424</td>
<td>Distributed computing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CA-1310</td>
<td>3 D multimedia &amp; Animation</td>
<td>3-0-0</td>
<td>3</td>
</tr>
</tbody>
</table>
To introduce the students the basic of C and Logic behind the implementation of different features of C like different data types, function, array, control statements, pointers, structures, file processing and recursion.

1. **OVERVIEW OF C PROGRAMMING & CONTROL STATEMENTS**
   - Introduction to C, Structure of C program, C character Set, Identifiers and keywords, data types, constants, variables and arrays, Declarations, expressions, statements, Symbolic constants, type conversion, operators and their hierarchy & associativity, Input and output functions in C, some simple C programs, header files, common programming errors
   - Control statements: Sequencing, Selection: conditions; the if statement; if statements with compound statements; nested if statements and multiple-alternative decisions; the switch statement. Loop Control Statements – For loop, while loop, Do-while loop and nested loops, break statement, The continue statement,

2. **TOP-DOWN DESIGN WITH FUNCTION:** Building programs from existing information; library functions; top-down design and structure charts; Defining a function, accessing a function, function prototypes, passing arguments to a function, Call by value and reference, Types of Storage Classes: Scope of variables – Global, auto and Static Variables, Recursion,

3. **ARRAYS & STRING HANDLING:** declaring, referencing and initializing arrays; array subscripts; using for loops for sequential access; using array elements as function arguments; array arguments; searching and sorting an array; multidimensional arrays; string basics; string library functions; assignment and substrings; longer strings; concatenation and whole line input; string comparison; character operations; string to number and number to string conversions.

4. **Pointers Basics**
   - Fundamentals, Pointer declarations, Passing pointers to the functions, pointers and one dimensional array, dynamic memory allocation, Operations on pointers, arrays of pointers. Function returning a Pointer, A pointer to a Function, pointers to pointers, pointers and strings, Void pointers. arguments to function main
5. **STRUCTURE, UNION AND PREPROCESSING:** Declaration and initialization of structure, structure within structure, Array of structure, Passing to a function, Pointer to structure, Union, Union of structure, Enumerated Data Type, typedef, Bitwise operators, Bit fields in Structures, Pre-processor directives, conditional compilation; defining macros with parameters

6. **FILE PROCESSING:** Introduction, Streams and File types, Opening and closing a data file, Input / Output operation on files, Text mode Versus Binary Mode, Formatted Input and output operation with files, Structures read and write in files, Random access to Files, Error detection during File operations, other useful file handling operations.

7. **INTRODUCTION TO ELEMENTARY DATA STRUCTURE:** Definition of data structures and abstract data types; linear vs. non-linear Data structure; stack, queue, linked list, static Vs Dynamic Implementation, static implementation of stack & queue

**TEXT BOOK**


**REFERENCE BOOKS**

4. E.Balagurusamy “C – programming” Tata McGray Hill
OBJECTIVE
To lay mathematical foundation for the fundamentals of various computational structures such as Boolean algebra, propositional logic, graph and trees.

PRE-REQUISITES
Knowledge of Data Structure

1. **SET THEORY**: Introduction to set theory; set operations; algebra of sets: duality, finite and infinite sets, classes of sets, power sets, multi sets, Cartesian product, representation of relations, types of relation, equivalence relations and partitions, partial ordering relations and lattices; function and its types, composition of function and relations; cardinality and inverse relations

2. **PROPOSITIONAL CALCULUS**: Basic operations: AND (Λ), OR (V), NOT (~), truth value of a compound statement, propositions, tautologies, contradictions.

3. **TECHNIQUES OF COUNTING**: Permutations with and without repetition, combination.

4. **RECURSION AND RECURRENCE RELATION**: Polynomials and their evaluation; sequences, introduction to AP, GP and AG series, partial fractions; linear recurrence relation with constant coefficients; homogeneous solutions, particular solutions, total solution of a recurrence relation using generating functions.

5. **ALGEBRiC STRUCTURES**: Definition and examples of a monoid, semigroup, groups and rings; homomorphism, isomorphism and automorphism; subgroups and normal subgroups; cyclic groups, integral domain and fields; co-sets; Lagrange’s theorem

6. **GRAPHS**: Introduction to graphs, directed and undirected graphs; homomorphic and isomorphic graphs; subgraphs; cut points and bridges; multigraph and weighted graph; paths and circuits, shortest path in weighted graphs; Eulerian path and circuits, Hamilton paths and circuits; planar graphs; Euler’s formula.

7. **TREES**: Trees, spanning trees, binary trees and its traversals

REFERENCE BOOKS
OBJECTIVE
To provide knowledge about various organizations and management information systems, keeping in view the aspects of shareability, availability, evolvability and integrity.

PRE-REQUISITES
Knowledge of data structures, discrete mathematical structures

1. INTRODUCTION: Purpose of Database system; Characteristics of database approach; Advantages of using DBMS; Database concept and architecture; Data Abstraction; Data Models; Instances and schema; Data independence; schema architecture; Database Languages; Database Manager; Database Administrator; Database Users.

2. DATA MODELING: Entity sets attributes and keys; Relationships (ER); Database modeling using entity; Type role and structural constraints; Weak and Strong entity types; Enhanced entity-relationship (EER); Entity-Relationship Diagram Design of an E-R Database schema; Object modeling; Specialization and generalization; Modeling of union types.

3. DATA MODELS AND RELATIONAL ALGEBRA: Introduction to Hierarchical model and Network model. Relational model -basic concepts; Enforcing Data Integrity Constraints; Relational-Algebra Operations; Assertion and Triggers; Introduction on views; Codd's Rules.

4. NORMALIZATION: Database design process; Relational database design; Relation Schema; Anomalies in a database ; Functional dependencies; Membership and minimal covers; Normal forms; Multi-valued dependencies; Join dependencies; Inclusion dependencies ; Reduction of an E-R schema to Tables; Converting EER Diagrams to relations.

5. QUERY LANGUAGES: Query-by-example(QBE); Introduction to SQL; Basic queries in SQL; Advanced queries in SQL; Functions in SQL; Basic data retrieval; Aggregation; Categorization; Updates in SQLs, Views in SQL.

6. FILE ORGANIZATION: Indexing and Hashing; Overview of file organization techniques; Secondary storage devices; Operations in files; Heap files and sorted files; Indexing and Hashing- Basic concepts; Static Hashing; Dynamic Hashing.

7. TRANSACTION PROCESSING: Desirable properties of transactions; Implementation of atomicity and durability; Concurrent executions; Schedules and recoverability; Serializability of schedules concurrency control; Concurrency Control; Deadlock handling - detection and resolution.

REFERENCE BOOKS

MCA-103 RELATIONAL DBMS

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
MCA

<table>
<thead>
<tr>
<th>MCA-104</th>
<th>DATA COMMUNICATIONS AND NETWORKING</th>
<th>L T P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3 0 0</td>
<td>3</td>
</tr>
</tbody>
</table>

**OBJECTIVE**
To have a fundamental understanding of the design, performance and state of the art of wireless communication systems, Topics covered include state of the art wireless standards and research and thus changes substantially form one offering of this course to the next

**PRE-REQUISITES**
Knowledge of computers hardware and software

4. **Internet Suite of Protocols: Internet and transport layer:** Introduction to Internet, Internet architecture, Internet Layer: IP Protocol, IP datagram, IP addressing and protocols, ARP, RARP, ICMP, IGMP. IP routing and protocols: RIP, OSPF, and BGP. Transport layer Protocols: TCP functions, segments and connections. UDP, TCP verses UDP.
5. **Network Layer:** Point - to - Pont Networks, routing, IP packet, IP address, IPv4 & IPv6. Congestion Control & Quality of services (QoS) – Congestion Control in TCP & Frame Relay Network; QOS; Flow Characteristics; Technique to improve Congestion Control; Scheduling; traffic shaping; QOS in Switched Network;

**REFERENCE BOOKS**
OBJECTIVE
To make students understand the concepts related to language development communication skills.

1. **FEATURES OF INDIAN ENGLISH:** Correction of sentences- structures- Tenses- ambiguity- idiomatic distortions.
2. **INFORMAL CONVERSATION Vs FORMAL EXPRESSION:** Verbal and non-verbal communication; barriers to effective communication- kinesics.
3. **TYPES OF COMMUNICATION:** Oral, Writing and Reading- Word- Power- Vocabulary- Jargon- rate of speech; pitch; tone- Clarity of voice.
4. **TECHNICAL PRESENTATION:** Types of presentation- video conferencing- participation in meetings- chairing sessions.
5. **FORMAL AND INFORMAL INTERVIEWS:** Ambiance and polemics- interviewing in different settings and for different purposes e.g. eliciting and giving information; recruiting; performance appraisal.
6. **WRITTEN COMMUNICATION:** Differences between spoken and written communication- features of effective writing such “as clarity; brevity; appropriate tone clarity; balance etc.
7. **LETTER-WRITING:** Business forma culture-style-effectiveness; promptness- Analysis of sample letters collected from industry- email; fax.

**REFERENCE BOOKS**
OBJECTIVE
To provide a foundation for numerical computing for scientific and engineering applications

PRE-REQUISITE
Knowledge of Basic Mathematics involving differentiation, integration, differential equations, linear equations, etc.

1. **ERRORS IN NUMERICAL CALCULATIONS:** Introduction; numbers and their accuracy; absolute; relative and percentage errors and their analysis; truncation errors; general formula; error calculation for inverse problem.

2. **SOLUTION OF NON-LINEAR EQUATIONS:** Bisection method; Regula-Falsi method; Secant method; Newton-Raphson method; fixed point method; initial approximation and convergence criteria.

3. **SOLUTION OF LINEAR SYSTEMS:** Gauss elimination method; Gauss-Jorden method; UV factorization, Jacobi’s method; Gauss-Seidal method.

4. **INTERPOLATION & CURVE FITTING:** Introduction to interpolation; Newton’s forward and backward formula; Sterling formula; Lagrangian polynomials; divided differences; least squares method.

5. **NUMERICAL DIFFERENTIATION AND INTEGRATION:** Derivatives from differences tables; numerical differentiation formulas, Newton-Cotes integration formulae; trapezoidal rule; Simpson’s rule; Bool’s rule; Weddle’s rule; Romberg’s rule.

6. **SOLUTION OF DIFFERENTIAL EQUATIONS:** Taylor’s series method; Euler and modified Euler’s method; Runge-Kutta method; Milne’s prediction corrector method, Adams–Bashforth method.

7. **SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS:** Finite difference approximation; solution of Laplace equation (standard 5 point formula) one-dimensional heat equation (Schmidt method, Cranck-Nicolson method; Dufort & Frankel method and wave equation.

REFERENCE BOOKS
1. Grewal B. S., “Numerical Methods in Engineering and Sciences”, Khanna Publisher
<table>
<thead>
<tr>
<th>MCA-105</th>
<th>OBJECT ORIENTED PROGRAMMING USING C++</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

**OBJECTIVE**
Providing a sound conceptual understanding of the fundamental concepts of computing hardware, software, networking and services; build programming logic and thereby developing skills in problem solving using C++ programming language; Introduce the concept of object orientation and on how to handle data in different forms; Emphasize the concepts and constructs rather than on language features.

1. **INTRODUCTION TO C++:** C++ standard library, basics of a typical C++ environment; pre-processors directives; illustrative simple C++ programs; header files and namespaces, library files.
2. **OBJECT ORIENTED CONCEPTS:** Introduction to objects and object oriented programming; encapsulation (information hiding); access modifiers: controlling access to a class; method, or variable (public, protected, private, package); other modifiers; Polymorphism: overloading, inheritance, overriding methods, abstract classes, reusability, class’ behaviors.
3. **CLASSES AND DATA ABSTRACTION:** Introduction; structure definitions; accessing members of structures; class scope and accessing class members; separating interface from implementation; controlling access function and utility functions, initializing class objects: constructors, using default arguments with constructors; using destructors; classes : const(constant) object and const member functions, object as member of classes, friend function and friend classes; using this pointer, dynamic memory allocation with new and delete; static class members; container classes and integrators; proxy classes; function overloading.
4. **OPERATOR OVERLOADING:** Introduction; fundamentals of operator overloading; restrictions on operators overloading; operator functions as class members vs. as friend functions; overloading, <<; >> overloading unary operators; overloading binary operators.
5. **INHERITANCE, VIRTUAL FUNCTIONS AND POLYMORPHISM:** Introduction, inheritance: base classes and derived classes, protected members; casting base-class pointers to derived-class pointers; using member functions; overriding base-class members in a derived class; public, protected and private inheritance; using constructors and destructors in derived classes; implicit derived-class object to base-class object conversion; composition vs. inheritance; virtual functions; abstract base classes and concrete classes; polymorphism; new classes and dynamic binding; virtual destructors; polymorphism; dynamic binding.
6. **FILES AND I/O STREAMS:** Files and streams; creating a sequential access file; reading data from a sequential access file; updating sequential access files, random access files; creating a random access file; writing data randomly to a random access file; reading data sequentially from a random access file; stream input/output classes and objects; stream output; stream input; unformatted I/O (with read and write); stream manipulators; stream format states; stream error states.
7. **TEMPLATES & EXCEPTION HANDLING:** Function templates; overloading template functions; class template; class templates and non-type parameters; templates and inheritance; templates and friends; templates and static members; basics of C++ exception handling: try, throw, catch, throwing an exception, catching
an exception, re-throwing an exception, exception specifications, processing unexpected exceptions; stack unwinding; constructors, destructors and exception handling; exceptions and inheritance.

REFERENCE BOOKS
MCA

<table>
<thead>
<tr>
<th>MCA-106</th>
<th>WEB DEVELOPMENT</th>
<th>L T P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3 1 0</td>
<td>4</td>
</tr>
</tbody>
</table>

OBJECTIVE
To impart knowledge of basic semesters of Internet, various standards like HTML, XML etc., client side and server side programming.

PRE-REQUISITES
Knowledge of Web Designing and Computer Network

1. INTRODUCTION TO THE INTERNET, THE WORLD WIDE WEB: The idea of hypertext and hyper media; how the web works: HTTP, HTML and URLs; how the browser works: MIME types, plug-ins and helper applications; standards: HTML, XML, XHTML and the W3C; functionality of MacroMedia DreamWeaver.

2. HYPERTEXT MARKUP LANGUAGE: The anatomy of an HTML document; marking up for structure and style: basic page markup, absolute and relative links, ordered and unordered lists, embedding images and controlling appearance, table creation and use, frames, nesting and targeting; descriptive markup: meta tags for common tasks, semantic tags for aiding search, the doubling code and RDF.

3. SEPARATING STYLE FROM STRUCTURE WITH STYLE SHEETS: Internal style specifications within HTML; external linked style specification using CSS, page and site design considerations.

4. CLIENT SIDE PROGRAMMING: Introduction to JavaScript syntax; JavaScript object model, event handling; output in JavaScript; Forms handling; miscellaneous topics such as cookies, hidden fields and images; applications.

5. SERVER SIDE PROGRAMMING: Introduction to server side technologies: ASP/JSP, programming languages for server side scripting; configuring the server to support ASP/JSP; applications; input/output operations on the WWW; forms processing, (using VBScript/JavaScript)

6. OTHER DYNAMIC CONTENT TECHNOLOGIES: Introduction to ASP & JSP, Delivering multimedia over web pages; the VRML idea; the Java phenomenon: applets and servelets; issues and web development.

7. INTRODUCTION TO MICROSOFT .NET TECHNOLOGY AND ITS COMPARISON WITH THE COMPETING TECHNOLOGIES.

REFERENCE BOOKS
OBJECTIVE
To provide basic knowledge of internals of microprocessor, its architecture, components, terminologies, etc. at minute level and ultimately about the working of a digital computer hardware as a whole

PRE-REQUISITES
Knowledge of data structures, microprocessors and interfacing

1. **INTRODUCTION TO DIGITAL COMPUTER**: Functions and Block Diagram of Computer, Types of Software – System software / Application software / Utility Software. Compilers, Interpreters, Assemblers, Linker, Loader & Programming Language Paradigm

2. **DATA REPRESENTATION AND BOOLEAN ALGEBRA**: Binary, Octal, HEX and their inter-conversion, 1’s and 2’s complement. Binary Arithmetic, Number Systems – BCD, EBCDIC, ASCII, De-Morgan’s Theorem, Duality Theorem, Boolean Algebra Rules, Postulates/Laws, Logic Circuits, NOT, AND, OR, NAND, NOR, XOR, XNOR.

3. **COMBINATIONAL & SEQUENTIAL CIRCUITS**: Half Adder, Full Adder, Binary Adder and Subtractor, Decoder / Encoder, Multiplexer / Demultiplexer, Sequential Circuits: Flip Flops - SR, D, JK, Master – Slave, Edge Triggered, Shift Registers, Counters: Synchronous as well as Asynchronous Counter

4. **MEMORY SYSTEM**: Memory Hierarchy, Primary Memory – DRAM, SDRAM, DDR, RDRAM. ROM, PROM, EPROM, EEPROM, Concepts of Auxiliary, Associative, Cache and Virtual Memory, DMA, DMA Transfer modes should be covered.


7. **MULTI-PROCESSOR ORGANIZATION**: Parallel Processing, Concept and Block Diagram, Types (SISD, SIMD, MIMD, MISD), Future Directions for Parallel Processors, Performance of Processors.

**REFERENCE BOOKS**

1. Computer Organization & Architecture, Carpinell
2. Computer System Architecture, Morris Mano
3. Advanced Computer Architecture, Kaith Hwang
4. Digital Computer Electronics, Malvino
7. Digital Electronics, Bartee
9. Intel Micro Processors, Barry Brey
OBJECTIVE
To equip students with computer based accounting and other financial skills.

1. **ACCOUNTING**: Principles; concepts; conventions; double entry system of accounting; introduction of basic books of accounts ledgers.
2. **PREPARATION OF TRIAL BALANCE**: Final accounts- company final accounts.
3. **FINANCIAL MANAGEMENT**: Meaning and scope; role; objectives of time value of money- over vitalization- under capitalization- profit maximization- EPS maximization.
4. **RATIO ANALYSIS**: Advantages; limitations; fund flow analysis- meaning; importance; preparation and interpretation of funds flow and cash flow statements- statement of changes in working capital.
5. **COSTING**: Nature and importance and basic principles. Absorption costing vs. marginal costing- Financial accounting vs. cost accounting vs. management accounting.
6. **MARGINAL COSTING AND BREAK-EVEN ANALYSIS**: Nature; scope and importance- practical applications of marginal costing; limitations and importance of cost- volume; profit analysis.
7. **STANDARD COSTING AND BUDGETING**: Nature; scope; computation and analysis- materials variance; labor variance and sales variance- budgeting- cash budget; sales budget- flexible Budgets; master budgets.

**REFERENCE BOOKS**
OBJECTIVE
To relay the theoretical and practical fundamental knowledge of most commonly used algorithms and data structure

PRE-REQUISITES
Knowledge of basic computer programming

SYLLABUS

1. INTRODUCTION TO DATA STRUCTURES: Definition of data structures and abstract data types; linear vs. non-linear data types; Static and Dynamic implementations; Arrays; 2; 3 and multi-dimensional arrays; Examples and real life applications.

2. TIMING COMPLEXITY: Time Complexity; Asymptotic Notations; Running Times; Best Case; Worst Case; Average Case; Introduction to Recursion; Divide and Conquer Algorithm; Recurrence Relation.

3. STACKS AND QUEUES: The Stacks: Definition; Array based implementation of stacks; Examples: Infix; postfix; prefix representation; Conversions; Applications; definition of Queues; Array based implementation of Queues, Towers of Hanoi Problem.

4. LINKED LISTS: Lists; Linked List implementation of stacks and queues; Circular implementation of Queues and Singly linked Lists; Straight / circular implementation of doubly linked Queues; Priority Queues; Applications, Josephus problem.

5. TREES: Definition of trees and Binary trees; Properties of Binary trees and Implementation; Binary Traversal pre-order; post order; In-order traversal; Binary Search Trees; Implementations; Threaded trees; Balanced multi way search trees; AVL Trees.

6. GRAPHS: Definition of Undirected and Directed Graphs and Networks; The Array based implementation of graphs; Adjacency matrix; path matrix implementation; The Linked List representation of graphs; Shortest path Algorithm; Graph Traversal – Breadth first Traversal; Depth first Traversal; Hash Tables; Hash function.

7. SORTING SEARCHING AND INDEXING: Introduction; Sorting by exchange; selection; insertions; bubble sort; Straight selection sort; Efficiency of above algorithms; Shell sort; Merge sort; Quick sort; Heap sort, Searching Algorithms: Straight Sequential Search; Binary Search (recursive & non–recursive Algorithms), Indexing.

REFERENCE BOOKS

WEB REFERENCES
OBJECTIVE
To cover the entire concept behind the cellular technology, including, the standards like GSM; CDMA and various design parameters for wireless system. Going through these topics will help the students to face telecom sector and software companies.

1. **INTRODUCTION TO WIRELESS COMMUNICATION SYSTEMS**: Evolution of mobile radio communications; examples of wireless comm. systems; paging systems; Cordless telephone systems; comparison of various wireless systems.

2. **MODERN WIRELESS COMMUNICATION SYSTEMS**: Second generation cellular networks; third generation wireless networks; wireless in local loop; wireless local area networks; Blue tooth and Personal Area networks.

3. **INTRODUCTION TO CELLULAR MOBILE SYSTEMS**: Spectrum Allocation; basic Cellular Systems; performance Criteria; Operation of cellular systems; analog cellular systems; digital Cellular Systems.

4. **CELLULAR SYSTEM DESIGN FUNDAMENTALS**: Frequency Reuse; channel assignment strategies; handoff Strategies; Interference and system capacity; tracking and grade off service; improving coverage and capacity.

5. **MULTIPLE ACCESS TECHNIQUES FOR WIRELESS COMMUNICATION**: Introduction to Multiple Access; FDMA; TDMA; Spread Spectrum multiple Access; space division multiple access; packet ratio; capacity of a cellular systems.

6. **WIRELESS NETWORKING**: Difference between wireless and fixed telephone networks; development of wireless networks; fixed network transmission hierarchy; traffic routing in wireless networks; wireless data services; common channel signaling; ISDN (Integrated Services digital Networks); advanced intelligent networks.

7. **INTELLIGENT CELL CONCEPT AND APPLICATION**: Intelligent cell concept; applications of intelligent micro-cell Systems; in-Building Communication; CDMA cellular Radio Networks.

**REFERENCE BOOK**
LIST OF EXPERIMENTS/EXERCISES
1. Basic/Simple logic building
2. Handling mathematical data
3. Use of control structures
4. Use of Function
5. Handling mathematical problems
6. Array and Pointer
7. Searching and Sorting
8. String Manipulation
9. Use of Structure and Union
10. File handling

REFERENCE BOOKS
LIST OF EXPERIMENTS

1. Record operations:
   a) Creating a table.
   b) Add a record, delete a record, modify the record in the database.
   c) Generate queries.
   d) Generate the report; listing all the records of database in ascending order.
   e) Create table from a given table.
   f) Insert the data into the table interactively means by using & operator.
2. Delete data from given table.
   • Delete data from table based on the given condition.
   • Update the contents of the table.
   • Modify the structure of the table.
   • Delete complete structure of the table.
3. Create keys: Table with primary key; Table with foreign key; Not null and Unique Constraints; Table with Check and Default Constraints; Insert data in the tables created with constraints.
4. Use operators: Scalar operators, Group operators, Pattern matching operator.
5. Locking and Unlocking the table, using different modes like Exclusive and Share etc.
6. Create View and see the relationship with table it has been created from, and finally, drop the view.
7. Use command to save the already executed command
   • Change contents the most recently executed commands.
   • Delete any line of the command.
   • Use Pseudo Columns in the Table.
   • Run the saved Command.
   • Edit using Word Processor and Save the command of Program in the desired Drive.
8. Create cursor, fetch data and show application of cursor.
9. Create a function and use cursor in the function.
10. Create Procedure.
11. Create Package and use Procedure and function.
12. Create a trigger on a Table.
13. Perform various table operations: Delete the structure, use delete command with conditions, update records of the table with conditions, Alter structure of the table, add a new column into the table, Change size of the existing column in the table, etc.
14. Create the view from the table, combine using equi-join, retrieve data with left join, outer join and self join.
MCA

REFERENCE BOOKS

<table>
<thead>
<tr>
<th>MA-252</th>
<th>APPLIED NUMERICAL METHODS LAB</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

LIST OF EXPERIMENTS
1. To find the roots of non-linear equation using Bisection method.
2. To find the roots of non-linear equation using Secant method.
3. To find the roots of non-linear equation using Newton's method.
4. To solve the system of linear equations using Gauss-Elimination method.
5. To solve the system of linear equation using Gauss-Seidal iteration method.
6. To find the values of function at a particular point using Newton’s forward formula.
7. To find the values of function at a particular point using Newton’s backward formula.
8. To find the values of function at a particular point using Lagrange’s interpolation formula.
9. To integrate numerically using Trapezoidal rule.
10. To integrate numerically using Simpson’s rule.
11. To find the solution of o.d.e (ordinary differential equation) by Euler’s method.
12. To find the solution of o.d.e by Runge-Kutta method.
13. To find the numerical solution of Laplace equation.
14. To find the numerical solution of heat equation.
15. To find the numerical solution of wave equation.

REFERENCE BOOKS
LIST OF EXPERIMENTS/EXERCISES
1. Write a program for Basic/Simple logic building in C++
2. Write a program to implement function overloading
3. Write a program to implement the concept of class and object
4. Write a program to implement the concept of friend function
5. Write a program to implement the concept of static data member
6. Write a program using the concept of constructor & destructor
7. Write a program to Implement operator overloading
8. Write a program to Implement single inheritance
9. Write a program to Implement Multiple inheritance
10. Write a program to Implement Virtual function
11. Write a program to create, read & write sequential file
12. Write a program to create, read & write random access file
13. Write a program to implement function template
14. Write a program to implement class template
15. Write a program for exception handling

Reference book
LIST OF EXPERIMENTS
1. Design a web-page using (a) frames and tables, (b) using style sheets
2. Design a web-page showing the use of forms
3. Design a web-page to show different validation checking using Java script
5. Write a program to implement (a) a class, (b) different sorting algorithms.
6. Write a program to show the use of constructor
7. Write a program to do (a) matrix multiplication and create user defined interface.
8. Write a program to create user defined packages.
9. Write a program to create a file and perform different operations on a file.
10. Write a program to design a calculator.
11. Write a program to show the different layouts available in java.
12. Write a program to design an applet to (a) display a face, (b) analog clock.
13. Write a program to design an applet for the students detail entry.
14. Write a program to show the Client-Server communication
15. Develop a website for the college
16. Develop a website for a the newspaper agency

REFERENCE BOOKS
LIST OF EXPERIMENTS

1. Write a program for inputting an array of integer and to sort them using Bubble sort, Insertion sort, Selection sort, Merge sort, Quick sort, perform each sorting using functions.
2. Write a program for inputting an array of integer and to search a particular element by using Linear search and binary search, recursively and iteratively using functions.
3. Write the program for implementing stack using array.
4. Write a program for implementing two stack within one array.
5. Write a program for implementing simple queue and circular queue using array.
6. Write a program for implementing two stack using two different array.
7. Write the pseudo code and program for implementing two priority queues within one array.
8. Write the program for implementing two priority queues within one array.
9. Write the program for implementing stack and simple queue within one array.
10. Write the program for implementing Input restricted dequeue using array.
11. Write a program for implementing Output restricted Dequeue using array.
12. Write a program for implementing different operations on matrices.
13. Write a program for implementing grounded simply singly link list.
14. Write a program for implementing header simply singly link list.
15. Write a program for implementing grounded circular singly linked list.
16. Write a program for implementing Header circular singly linked list.
17. Write pseudo code and program for implementing grounded simply doubly linked list
18. Write a program for implementing grounded and header simply doubly linked list
19. Write a program for implementing grounded circular doubly linked list.
20. Write a program for implementing headed circular doubly linked list.
21. Write a program for implementing stacks using linked list.
22. Write a program for implementing queue using linked list.
23. Write a program for implementing binary search tree.
24. Write a program for implementing polynomial manipulation.
25. Write a program for inputting an array and to sort them using heap sort.
26. Write a program for implementing sparse matrices.
27. Write a program for converting infix expression to suffix.
OBJECTIVE
To provide the knowledge of internals, different types and purpose of operating systems

PRE-REQUISITES
Knowledge of computer organization and architecture, programming skills

1. **INTRODUCTION**: Introduction to operating system concepts (including multitasking, multiprogramming, multi user, multithreading, etc.); types of operating systems: batch operating system, time-sharing systems, distributed OS, network OS, real time OS, embedded and smart card OS; various operating system services, architecture, system programs and calls.

2. **PROCESS MANAGEMENT AND THREADS**: Process concept, Life cycle and implementation of process, Thread usage and implementation in user space and in kernel, process scheduling, operation on processes; CPU scheduling, scheduling criteria, scheduling algorithms - First Come First Serve (FCFS), Shortest-Job-First (SJF), priority scheduling, Round Robin (RR), multilevel queue scheduling.

3. **MEMORY MANAGEMENT**: Logical & physical address space; swapping; contiguous memory allocation, non-contiguous memory allocation paging and segmentation techniques, segmentation with paging; virtual memory management - demand paging & page-replacement algorithms; demand segmentation.

4. **FILE SYSTEM**: Different types of files and their access methods, directory structures; various allocation methods; disk scheduling and management and its associated algorithms; introduction to distributed file system.

5. **PROCESS-SYNCHRONIZATION & DEADLOCKS**: Critical section problems, mutual exclusion with busy waiting, semaphores; methods for handling deadlocks: deadlock prevention, avoidance and detection; deadlock recovery; Classical IPC problems: dining philosophers’ problem, readers-writers problem.

6. **I/O SYSTEMS**: I/O hardware, device controllers, interrupt handlers, device drivers, application I/O interface, kernel, transforming I/O requests, performance issues.

7. **LINUX/UNIX SYSTEM**: LINUX/UNIX architecture; UNIX system calls for processes and file system management; basic commands of LINUX/UNIX; shell interpreter, shell scripts.

**REFERENCE BOOKS**
OBJECTIVE
Students completing this course are expected to be able to:
- Write programs that utilize the OpenGL graphics environment.
- Use polygonal and other modeling methods to describe scenes.
- Understand and be able to apply geometric transformations.
- Create basic animations.
- Understand scan-line, ray-tracing, and radiosity rendering methods

PRE-REQUISITES
Knowledge of computer programming, 2D and 3D geometry

1. INTRODUCTION: What is computer graphics, computer graphics applications, computer graphics hardware and software, two dimensional graphics primitives: points and lines, line drawing algorithms: DDA, Bresenham’s; circle drawing algorithms: using polar coordinates, Bresenham’s circle drawing, mid point circle drawing algorithm; polygon filling algorithm, boundary filled algorithm, scan-line algorithm, flood fill algorithm.
2. TWO DIMENSIONAL VIEWING: The 2-D viewing pipeline, windows, viewports, window to view port mapping; clipping: point, clipping line (algorithms): 4 bit code algorithm, Sutherland-Cohen algorithm, parametric line clipping algorithm (Cyrus Beck).
3. POLYGON CLIPPING ALGORITHM: Sutherland-Hodgeman polygon clipping algorithm, homogeneous coordinates system, two dimensional transformations: transformations, translation, scaling, rotation, reflection, shearing, transformation, composite transformation.
4. THREE DIMENSIONAL GRAPHICS: Three dimensional graphics concept, matrix representation of 3-D transformations, composition of 3-D transformation; viewing in 3D: projections, types of projections; the mathematics of planner geometric projections; coordinate systems.
5. HIDDEN SURFACE REMOVAL: Introduction to hidden surface removal; the Z-buffer algorithm, scan-line algorithm, area sub-division algorithm.
6. REPRESENTING CURVES AND SURFACES: Parametric representation of curves: Bezier curves, B-Spline curves; parametric representation of surfaces; interpolation method.
7. ILLUMINATION, SHADING, IMAGE MANIPULATION: Illumination models, shading models for polygons, shadows, transparency; what is an image, filtering, image processing, geometric transformation of images.

REFERENCE BOOKS
OBJECTIVE
To relay the theoretical and practical aspects of design of algorithms

PRE-REQUISITES
Knowledge of fundamentals of basic computer programming for implementing algorithms.

1. MATHEMATICAL PRELIMINARIES: Review of growth functions; Solution of difference equations; Analysis in terms of space and time complexity; Insertion sort; Merge sort; Heap sort; Quick sort.
2. ADVANCED DATA STRUCTURES: binary search trees; red black trees; binomial heap; Priority queues
3. DYNAMIC PROGRAMMING: Matrix multiplications; longest common subsequence; 0/1 knapsack; the traveling salesperson problem.
4. GREEDY ALGORITHMS: Activity selection; Huffman coding; task scheduling problem.
5. Elementary graph algorithm: Representation of Graph; Breadth-first search; Depth first search; topological sort; strongly connected components
7. Back Tracking: General method; 8 queens’ problem; graph colouring; Hamiltonian cycles; analysis of these problems. Introduction to NP hard & NP complete

TEXT BOOKS

REFERENCES:
OBJECTIVE
To provide basic knowledge of properties of software and its development processes, software quality, CASE tools, etc.

PRE-REQUISITES
Knowledge of computer programming, principles of management

SYLLABUS

1. INTRODUCTION: Definition and emergence of software engineering; evolving role of software; software life cycle models; Software Characteristics; Applications; Software Product; Software Process; Software crisis; software myths.

2. SOFTWARE PROJECT MANAGEMENT: Project management concepts; software process and project metrics Project planning; project size estimation metrics; project estimation Techniques; empirical estimation techniques; COCOMO- A Heuristic estimation techniques; staffing level estimation; team structures; risk analysis and management; project scheduling and tracking.

3. REQUIREMENTS ANALYSIS AND SPECIFICATION: requirements engineering; system modeling and simulation; Analysis principles modeling; partitioning Software; Prototyping: methods and tools; Specification principles; Representation; the software requirements specification and reviews Analysis Modeling: Data Modeling; Functional modeling and information flow: Data flow diagrams; Behavioral Modeling; The mechanics of structured analysis: Creating entity/ relationship diagram; data flow model; control flow model; the control and process specification; The data dictionary; Other classical analysis methods.

4. SYSTEM DESIGN: The design process: design principles; Design concepts, Functional independence; Cohesion; Coupling; Design documentation; Architectural Design: Software architecture; Data Design: Architectural Styles; Transform mapping and Transaction mapping.

5. TESTING AND MAINTENANCE: Software Testing Techniques; software testing fundamentals: objectives; principles; testability; Test case design; white box testing; basis path testing; Control structure testing: Black box testing; testing for specialized environments; architectures and applications. Software Testing Strategies: Verification and validation; Unit testing; Integration testing; Validation testing; the art of debugging; the debugging process debugging approaches. Software re-engineering; reverse engineering; Software Configuration Management.

6. SOFTWARE RELIABILITY AND QUALITY ASSURANCE: Quality concepts; Software quality assurance; SQA activities; Software reviews: formal technical reviews: Formal approaches to SQA; Statistical software quality assurance;
The ISO 9000 Quality standards; The ISO 9001 and Six Sigma standards; software reliability: Measures of reliability and availability.

7. **COMPUTER AIDED AND OBJECT ORIENTED SOFTWARE ENGINEERING**: CASE; building blocks; integrated case environments and architecture.

**REFERENCE BOOKS**


**WEB REFERENCES**

OBJECTIVE
To introduce to the students the in-depths of UNIX operating system structure and function, as well as to acquaint them with programming using Shell commands, and handling advanced concepts like semaphores.

1. **UNIX UTILITIES**: introduction to UNIX file system; vi editor; file handling utilities; security by file permissions; process utilities; disk utilities; networking commands; cp; mv; ln; rm; unlink; mkdir; rmdir; du; df; mount; umount; find; umask; ulimit; ps; who; w; finger; arp; ftp; telnet; rlogin; text processing utilities and backup utilities; detailed commands to be covered are cat; tail; head; sort; nl; uniq; ggrep; egrep; fgrep; cut; paste; join; tee; pg; comm.; cmp; diff; tr; awk; tar; cpio.

2. **PROBLEM SOLVING APPROACHES IN UNIX**: Using single commands; using compound commands; shell scripts; C programs; building own command library of programs; working with the Bourne shell: what is a shell; shell responsibilities; pipes and input redirection; output redirection; here documents; the shell as a programming language; shell meta character; shell variables; shell commands; the environment; control structures; shell script examples.

3. **UNIX FILES**: UNIX file structure; directories; files and devices; system calls; library functions; low level file access; usage of open; creat; read; write; close; lseek; stat; fstat; octl; umask; dup; dup2; the standard I/O (fopen; fclose; fflush; fseek; fgetc; getchar; fputc; putc; putchar; fgets; gets); formatted I/O; stream errors; streams and file descriptors; file and directory maintenance (chmod; chown; unlink; link; symlink; mkdir; rmdir; chdir; getcwd); directory handling system calls (opendir; readdir; closedir; rewinddir; seekdir; telldir).

4. **UNIX PROCESS AND SIGNALS**: what is process; process structure; starting new process; waiting for a process; zombie process; process control; process identifiers; system call interface for process management-fork; vfork; exit; wait; waitpid; exec; system; Signals: Signal functions; unreliable signals; interrupted system calls; kill and raise functions; alarm; pause functions; abort; sleep functions.

5. **INTERPROCESS COMMUNICATION OVERVIEW**: introduction to IPC; IPC between processes on a single computer system; IPC between process on different systems; file and record locking; other UNIX locking techniques; pipes; FIFO; streams and messages; namespaces; introduction to three types of IPC(System-V)-message queues; semaphores and shared memory.

6. **MESSAGE QUEUES**: UNIX System-V messages; UNIX kernel support for message; UNIX APIs for messages client/server example.

7. **SEMAPHORES**: UNIX System-V semaphores; UNIX kernel support for semaphores; UNIX APIs for semaphores; file locking with semaphores; Shared Memory- UNIX System-V shared memory; UNIX kernel support for shared memory UNIX APIs for shared memory; semaphore and shared memory example.

**REFERENCE BOOKS**
1. W. R. Stevens, “Unix Network Programming”, Pearson Education/Prentice Hall of India
OBJECTIVE
To relay the theoretical and practical knowledge of Advanced Java programming language

PRE-REQUISITES
Basic Knowledge of programming language and object oriented programming

1. **CORE JAVA**: Introduction to Java; Data types; variables; operators; Arrays; Control Statements; Classes & Methods; Inheritance; Exception Handling; Multithreading; Collections; I/O streams; AWT & Applet Programming; Swings.

2. **NETWORK PROGRAMMING**: Networking basics; Socket; port; Proxy servers; Internet addressing and URL; java.net –networking classes and interfaces; Implementing TCP/IP based Server and Client. Classes to be covered Socket; ServerSocket; IP Address; URL connections

3. **JAVA DATABASE CONNECTIVITY**: Types of JDBC Drivers; Writing JDBC applications using select; insert; delete; update; Types of Statement objects (Statement; PreparedStatement and CallableStatement); ResultSet; ResultsetMetaData; Inserting and updating records.

4. **REMOTE METHOD INVOCATION AND JAVA BEANS**: Introduction of RMI & Architecture; Implementing RMI Methods; Introduction to Java Bean; Rules for writing a Simple Bean; Using Beans to Build an Application.

5. **SERVLETS**: Configuring directory structure for a web application; Servlet API Overview; Writing and running Simple Servlet. Servlet Life Cycle; GenericServlet and HttpServlet; ServletConfig & ServletContext; Writing servlet to Handle Get and Post Methods; Reading user request data; Concept of cookie; Reading and writing cookies

6. **JAVA SERVER PAGES**: Why JSP? JSP Directives; writing simple JSP page; Scripting Elements; JSP Actions: JSP & Java Beans; JSP Actions: include; forward and plugin; Managing sessions using JSP; JSP & Databases; Error Handling in JSP; Writing custom tags; Different scopes in a JSP page; Using JDBC in JSP; Study and Development of a Web Application and an Assignment.

7. **INTRODUCTION TO STRUCTS**: A Web Application Framework – struts-config.xml; Understanding MVC architecture; Action\Servlet; Action\Form; Action\Mapping; Action classes.

REFERENCE BOOKS
7. O’Reilly, “Servlet and JSP”.
OBJECTIVE
To provide basic knowledge of image compression, audio, video, sound, virtual reality, intelligent multimedia systems etc

PRE-REQUISITES
Knowledge of computer graphics, programming, 3D geometry

1. **BASICS OF MULTIMEDIA TECHNOLOGY:** Computers; communication and entertainment; multimedia an introduction; framework for multimedia systems; multimedia devices; CD- Audio; CD-ROM; CD-I; presentation devices and the user interface; multimedia presentation and authoring; professional development tools;

2. **LAN AND MULTIMEDIA:** internet; World Wide Web & multimedia distribution network: ATM & ADSL; multimedia servers & databases; vector graphics; 3D graphics programs; animation techniques; shading; anti aliasing; morphing; video on demand.

3. **IMAGE COMPRESSION & STANDARDS:** Making still images; editing and capturing images; scanning images; computer color models; color palettes; vector drawing; 3D drawing and rendering; JPEG: objectives and architecture; DCT encoding and quantization; statistical coding; predictive lossless coding; performance; overview of other image file formats as GIF; TIFF; BMP; PNG; etc.

4. **AUDIO:** Digital representation of sound; time domain sampled representation; method of encoding the analog signals; sub-band coding; Fourier method; transmission of digital sound; digital audio signal processing; stereophonic and quadraphonic signal processing; editing sampled sound; MPEG Audio; audio compression and decompression; brief survey of speech recognition and generation; audio synthesis; musical instrument digital interface

5. **VIDEO:** digital video and image compression; MPEG motion video compression standard; DVI technology; time base media representation and delivery.

6. **VIRTUAL REALITY:** Applications of multimedia; intelligent multimedia system; desktop virtual reality; VR operating system; virtual environment displays and orientation making; visually coupled system requirements; intelligent VR software systems.

7. **MULTIMEDIA INFORMATION RETRIEVAL.**

**REFERENCE BOOKS**
OBJECTIVE
The course on RAD focuses on building applications within a very short time period. After successful completion of this course the students will be able to obtain a firm foundation on RAD concepts and methodologies and acquire sufficient working knowledge in RAD tools.

PRE-REQUISITES
Knowledge of programming, object oriented Concept and Database.

Unit-1: Introduction To Rapid Application Development: Definition, history; effect of mistakes on development schedule; importance; modern rapid life cycle model; modern RAD.

Unit-2: Issue In Rapid Application Development: Characteristics; strategy; constraints; advantages and disadvantages; customer oriented development; different RAD tools: open source versus licensed software builder; Easy Eclipse/Net-Beans/Anjuta/Glade, Visual Studio/Net, etc.


Unit-5: Reading keystrokes, handling mouse, creating menus, toolbars, buttons, status bar prompts, dialog box, check box, radio buttons, list boxes, combo boxes, sliders, multiple documents.

Unit-6: DLL’s, OLE Object Technologies, Creating Internet Programs using Visual C++ and Visual Basic, Creating Active X Controls, connecting to Database (using DAO/ADO/RDO) using Visual Basic and Visual C++.
MCA

Unit-7: Serialization, file handling, debugging.

Text Books

2. Microsoft Visual C++ By Steven Holzner (Pub: BPB)
3. Visual C++ Programming, 2\textsuperscript{nd} edition by Steven Holzner(Pub: PHI)

Reference Books

2. Using Visual Basic for Applications By Paul Sanna
3. Visual Basic Tech yourself in 21 Days (Tech Media Publication)
4. Visual Basic Programming By Steven Holzner
OBJECTIVE
To equip students with C# programming Concepts

1. **PHILOSOPHY OF .NET:** overview of distributed computing; origin of .NET technology; understanding; .NET platform; do's and don'ts of .NET; benefits and limitations of .NET; building blocks of .NET framework; .NET programming languages; role of MSIL and Metadata; .NET types and .NET namespaces.

2. **VISUAL STUDIO .NET AND ITS MAJOR COMPONENTS:** understanding CLR; CTS and CLS; developing C# Applications using Visual Studio .Net

3. **EVOLUTION OF C#:** comparison among C++; Java and C#; benefits of C#; object-oriented programming using C#

4. **C# PROGRAMMING:** introduction to C#; creating a C# program; types in C#; classes; inheritance and polymorphism; methods; statements and control; arrays and strings; interfaces; abstract and base classes.

5. **STATEMENTS AND CONTROL:** properties and indexers; delegates and their usefulness; attributes; I/O in C#; exception and error handling in C#; C# and windows application.

6. **ADO .NET:** comparison of ADO and ADO.NET; introduction to data access with ADO.NET components of ADO.NET; overview of XML; XML and ADO.NET.

7. **WEB DEVELOPMENT AND ASP .NET:** comparison of ASP and ASP.NET; features of ASP.NET; benefits of ASP.NET; features provided by ASP.NET; web forms and their components; overview of web services.

**REFERENCE BOOKS**
3. Pappas & Murray, “C# Essentials”, Prentice Hall of India
5. Wakefield, “C# and .NET Web Developers Guide”, IDG Books India
OBJECTIVE
This course introduces basic concepts, tasks, methods, and techniques in data mining. The emphasis is on various data mining problems and their solutions. Students will develop an understanding of the data mining process and issues, learn various techniques for data mining, and apply the techniques in solving data mining problems using data mining tools and systems. Students will also be exposed to a sample of data mining applications.

PRE-REQUISITES
Basic knowledge of database management system

1. DATA WAREHOUSING: Definition, usage and trends. DBMS vs data warehouse; data marts; metadata; multidimensional data mode; data cubes; schemas for multidimensional database: stars, snowflakes and fact constellations.
2. DATA WAREHOUSE PROCESS AND ARCHITECTURE: OLTP vs OLAP, ROLAP vs MOLAP; types of OLAP, servers, 3-Tier data warehouse architecture; distributed and virtual data warehouses; data warehouse manager.
3. DATA WAREHOUSE IMPLEMENTATION: Computation of data cubes; modelling OLAP data, OLAP queries manager; data warehouse back end tools; complex aggregation at multiple granularities; tuning and testing of data warehouse.
4. DATA MINING: Definition and task; KDD versus data mining; data mining techniques, tools and applications.
5. DATA MINING QUERY LANGUAGES: Data specification, specifying knowledge; hierarchy specification; pattern presentation and visualization specification; data mining languages and standardization of data mining.
6. DATA MINING TECHNIQUES: Association rules; clustering techniques; decision tree knowledge discovery through neural networks and genetic algorithm; rough sets; support vector machines and fuzzy techniques.
7. MINING COMPLEX DATA OBJECTS: Spatial databases, multimedia databases, time series and sequence data; mining text databases and mining Word Wide Web.

REFERENCE BOOKS
2. Han Jiawei and Kamber Micheline, “Data Mining - Concepts & Techniques”, Morgan Kaufmann, 2001
OBJECTIVE
To provide the foundation required for becoming a good software project manager by means of planning, evaluation and estimation, risk management, allocation and monitoring of resources, controlling software quality

PRE-REQUISITES
Knowledge of software engineering and the basic principles of management.

1. **INTRODUCTION**: Definition of a Software Project (SP), SP vs. other types of projects activities covered by SPM; categorizing SPs; project as a system; management control, requirement specification; information and control in organization

2. **STEPWISE PROJECT PLANNING**: Introduction, selecting a project; identifying project scope and objectives; identifying project infrastructure, analyzing project characteristics; identifying project products and activities; estimate efforts each activity; identifying activity risk; allocate resources; review/ publicize plan

3. **PROJECT EVALUATION AND ESTIMATION**: Cost benefit analysis; cash flow forecasting; cost benefit evaluation techniques; risk evaluation; Selection of an appropriate project report; Choosing technologies, choice of process model, structured methods: rapid application development, water fall, V-process-, spiral- models; Prototyping; delivery; Albrecht function point analysis

4. **ACTIVITY PLANNING AND RISK MANAGEMENT**: Objectives of activity planning; project schedule; projects and activities; sequencing and scheduling activities, network planning model; representation of lagged activities; adding the time dimension, backward and forward pass; identifying critical path; activity throat, shortening project; precedence networks; Risk Management: Introduction, the nature of risk, managing risk, risk identification, risk analysis, reducing the risks, evaluating risks to the schedule, calculating the z values

5. **RESOURCE ALLOCATION AND MONITORING THE CONTROL**: Introduction, the nature of resources, identifying resource requirements; scheduling resources creating critical paths; counting the cost; being specific; publishing the resource schedule; cost schedules, the scheduling sequence; Monitoring the control: Introduction, creating the frame work, collecting the data, visualizing progress, cost monitoring, earned value, prioritizing monitoring, getting the project back to target, change control

6. **MANAGING CONTRACTS AND PEOPLE**: Introduction, types of contract, stages in contract, placement, typical semesters of a contract, contract management, acceptance, Managing people and organizing semesters: Introduction, understanding behavior, organizational behavior: a back ground, selecting the right person for the job, instruction in the best methods, motivation, working in groups, becoming a team, decision making, leadership, organizational structures, conclusion, further exercises

7. **SOFTWARE QUALITY**: Introduction; the place of software quality in project planning; the importance of software quality; defining software quality, ISO 9126; Practical software quality measures; product versus process quality management; external standards; techniques to help enhance software quality; Study of any software project
MCA

management software: viz Project 2005 or equivalent

REFERENCE BOOKS
OBJECTIVE
To equip students with knowledge about handling various IT related activities such as taking decisions under varied situations, handling of resources, etc.

1 **NATURE AND FUNCTIONS OF MANAGEMENT**: Importance of management; definition of management; management process; roles of manager; management - a science or art, management - a profession.

2 **PLANNING & DECISION MAKING**: Nature of planning; importance of planning; types of planning; steps on planning; meaning of decision, types of decisions; organization: span of management, principles of organizing, departmentalization.

3 **DIRECTION**: Requirements of effective direction; motivation; importance of communication; purposes of communication; formal communication; informal communication; barriers to communication; principles of effective communication.

4 **LEADERSHIP**: Difference between a leader and a manager; characteristics of leadership; functions of a leader; approaches to leadership; effective leadership; leadership style in indian organizations.

5 **MANAGERIAL CONTROL & SOCIAL RESPONSIBILITIES OF BUSINESS**: Steps in a control process, need for control, types of control methods, essentials of effective control systems; meaning of social responsibility & social responsibilities of business towards different groups.

6 **INTRODUCTION TO ICT**: Recent IS/ICT trends; ICT management; ICT architectures; principles and models of ICT management; Corporate Performance Management.

7 **STANDARDS AND STRATEGIES of ICT**: International IT management standards(ITIL,COBIT); ICT governance and performance reference model strategic management of ICT;ICT services management ; Financial management of ICT.

**REFERENCE BOOKS**
LIST OF EXPERIMENTS
1. 2D line as raster graphics display using Bresenhem line drawing algorithm
2. 2D line drawing as raster graphics display using DDA line drawing algorithm
3. Circle drawing as raster graphics display using mid point circle drawing algorithm
4. Polygon filling as raster graphics display using Boundary fill algorithm and Flood fill algorithm
5. Line clipping
6. Polygon clipping
7. Display 3D object as 2D raster graphics display using perspective transformation
8. Rotation for 3D object about arbitrary axis
9. Hidden surface removal from a 3D object
10. 2D transformations of a given object (triangle, rectangle, pentagon) for translating, scaling, rotating, reflecting, shearing
11. Create a screen saver using inbuilt functions of graphics
12. Zoom an object
13. Reverse zooming
14. Create a Bezier Curve

REFERENCE BOOKS
LIST OF EXPERIMENTS
1. Write a shell script to generate a multiplication table.
2. Write a shell script that copies multiple files to a directory.
3. Write a shell script that counts that number of lines and words present in a given file.
4. Write a shell script that displays the list of all files in the given directory.
5. Write a shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two division options: one returns the quotient and the other returns reminder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add (-a), subtract (-s), multiply (-m), quotient (-c) and reminder (-r).
6. Write a shell script to reverse the rows and columns of a matrix.
7. Write a C program that counts the number of blanks in a text file (a) Using standard I/O and (b) Using system calls.
8. Implement in C the following Unix commands using system calls: cat, ls, mv
9. Write a program that takes one or more file/directory names as command line input and reports the following information of the file:
   (a) File type, (b) Number of links, (c) Time of last access, (d) Read, Write and Execute permissions.
10. Write a C program that illustrates uses of the mkdir, opendir, readdir, closedir, and rmdir APIs.
11. Write a C program that illustrates how to execute two commands concurrently with a command pipe.
12. Write a C programs that illustrates Two-way communication using the following:
   (a) unidirectional pipes and (b) bidirectional pipes.
13. Write a C program that illustrates the creation of child process using fork system cell.
14. Write a C program that displays the real time of a day every 60 seconds.
15. Write a C program that illustrates file-locking using semaphores.
16. Write a C program that implements a producer-consumer system with two processes. (Using semaphores)
17. Write a C program that illustrates inter process communication using shared memory system calls.
18. Write a C program that illustrates the following:
   (a) Creating a message queue, (b) Writing to a message queue and (c) Reading from a message queue.

REFERENCE BOOKS
LIST OF EXPERIMENTS
1. Program for deleting a text file.
2. Program for making connection between client and server.
3. To display a four by four identity matrix.
4. Define a class fruit with the following attributes:
   a. Name of the fruit
   b. Single fruit or bunch fruit
   c. Price
   d. Define a suitable constructor and displayFruit() method that displays values of all the attributes. Write a program that creates 2 objects of fruit class and display their attributes.
5. Write a Java program that will print details about the current date, time, month year, day of the month, day of the week.
6. To write a Java program that will contain two arrays on containing the products and the other containing the prices and to display the same.
7. Write a java program to show the utility of package.
8. Create an User-defined exception class using the extends keyword. Write a constructor for this class that takes a string argument and stores it inside the object with a string handle. Write a method that prints out the stored string. Create a try-catch clause to exercise the created exception.
9. Create a Java program using thread.
10. Write a program to accept two names as command line parameters. Check whether each of them exist in c:\java directory. If it exists, display its name and size, else, display the message that it does not exist. Further, if the extension of the file is “html” then it has to be deleted.
11. Write a java program to accept text from the user until he types “end” in a newline. The name of the file to store the contents should be accepted from the user at the command line. If no command line parameters are provided, the program should print error message and exit. Ensure if the same file has been created by listing the contents of the same.
12. Create an Applet to display a string “I am in the centre” in Courier font, with size 30 and style and italic. This text should be centered both horizontally and vertically.
13. Create an applet to obtain the list of fonts available with the current Java working environment.
14. Write an animation applet that makes an image appear from left to right, in stages.
15. Create a simple, non-editable combo box with a list of items, when selected one of the items, will display the string to the console and also printing the string which is being deselected, i.e., the string which already been selected.
16. Write a program that takes as input the telephone bill amount for 12 months and calculate the average telephone bill for the year. The program should also display the least and the maximum bill amounts.
17. Write a program through which the insert statement can be given at runtime. Use the program to insert the following test data in the master and details tables.
18. Write a program to delete a particular row according to required information.
19. Write an Echoserver and echoclient program that displays whatever is typed in the server on to the client using sockets.
20. Use socket programming to design a client/server application that takes the password as input and checks whether it is correct. The program should print the appropriate message.
21. Using beans to build an application
22. Using servlet develop a java program(Database connectivity)
23. Using RMI develop a client, server frame

<table>
<thead>
<tr>
<th>MCA-255</th>
<th>MULTIMEDIA SYSTEMS LAB</th>
<th>L T P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0-1-2</td>
<td>2</td>
</tr>
</tbody>
</table>

**LIST OF EXPERIMENTS**

**FLASH**
1. To create motion tween using Macromedia Flash.
2. To create and show the use of guided layer in Macromedia Flash.
3. To perform shape tweening in Macromedia Flash.
4. To draw an animated pool table in Macromedia Flash.
5. To create animated bow and arrow hitting a ball in Macromedia Flash.
6. To illustrate masking in Macromedia Flash.
7. To write any Text with the illusion of pen writing letters. (Example: Alif Laila) in Macromedia Flash.
8. To implement any one example of morphing in Macromedia Flash.
9. To create a Custom Cursor in Macromedia Flash.
10. To create an Animated Button in Macromedia Flash.
11. To implement START and STOP button in Macromedia Flash.
12. To create a web page of your choice and include an appropriate animation also.

**BLENDER**
15. Create a candle with stand in blender.
16. Create a Pair of Dice in Blender.
17. Create a First Animated Model (Yellow Submarine in blender).
18. Create a character (Character Modeling) in Blender.
LIST OF EXPERIMENTS
1. Write a program in C# illustrating the use of sequence, conditional and iteration construct.
2. Write a program in C# illustrating various operators like logical, arithmetical, relational, etc.
3. Write a program in C# illustrating overloading of various operators.
4. Write a program in C# illustrating use of friend, inline and Static Member functions, default arguments.
5. Write a program in C# illustrating use of destructor and various types of constructor.
6. Write a program in C# illustrating various forms of inheritance.
7. Write a program in C# illustrating use of virtual functions, Virtual base class, delegates.
8. Write a program in C# illustrating file operations.
9. Write a program in C# illustrating simple web applications using ASP.net
10. Write a program in C# illustrating use of Active X Controls.
OBJECTIVE
To introduce about artificial intelligence approaches to problem solving, various issues involved and application areas

PRE-REQUISITES
Knowledge of neural networks, data structures

1. INTRODUCTION TO AI AND SEARCH TECHNIQUES: Foundation and history of AI; data, information and knowledge; AI problems and techniques – AI programming languages, problem space representation with examples; blind search strategies, breadth first search, depth first search, heuristic search techniques: hill climbing: best first search, A * algorithm AO* algorithm, Means-ends analysis.

2. KNOWLEDGE REPRESENTATION ISSUES: predicate logic; logic programming; constraint propagation; representing knowledge using rules.

3. REASONING UNDER UNCERTAINTY: Reasoning under uncertainty, non monotonic reasoning; review of probability; Bayes’ probabilistic interferences and Dempster Shafer theory; heuristic methods; symbolic reasoning under uncertainty; statistical reasoning, fuzzy reasoning.

4. PLANNING & GAME PLAYING: Minimax search procedure; goal stack planning; non linear planning, hierarchical planning, planning in situational calculus; representation for planning; partial order planning algorithm

5. LEARNING: Basic concepts; rote learning, learning by taking advices, learning by problem solving, learning from examples, discovery as learning, learning by analogy; explanation based learning; neural nets; genetic algorithms.

6. OTHER KNOWLEDGE STRUCTURES: semantic nets, partitioned nets, parallel implementation of semantic nets; frames, common sense reasoning and thematic role frames; architecture of knowledge based system; rule based systems; forward and backward chaining; frame based systems.

7. APPLICATIONS OF ARTIFICIAL INTELLIGENCE: Principles of natural language processing; rule based systems architecture; expert systems, knowledge acquisition concepts; AI application to robotics, and current trends in intelligent systems; parallel and distributed AI: psychological modeling, parallelism in reasoning systems, distributed reasoning systems and algorithms

REFERENCE BOOKS
MCA

MCA-302  SYSTEM & NETWORK ADMINISTRATION  L T P  Cr

3 1 0  4

OBJECTIVE
To lay a strong foundation for overall system and network management

PRE-REQUISITES
Knowledge of computer organization and architecture, operating system, computer networks

1. INTRODUCTION TO SYSTEMS AND NETWORK ADMINISTRATION: Scope of systems and network administration; goals of systems and network administration; system components and their management
2. OPERATING SYSTEMS UTILITIES: Windows and Unix variants; file systems and standards (UFS, NFS, NTFS); processes and job control; privileged, user and group accounts; logs and audits; advanced scanning concepts and tools; advanced sniffer.
3. HOST MANAGEMENT: Booting and shutting down of an operating system; formatting, partitioning and building a file system; file system layout; concept of swap space; OS installation; installation and configuration of devices and drivers
4. SERVER CONFIGURATION & TROUBLESHOOTING: Linux/Windows server configuration; superuser/ administrator privileges; user management, controlling user resources; disk space allocation and quotas; process management (monitoring, killing/stopping, monitoring activity); file system repair, backup and restoration; integrating multiple operating systems; system sharing; authentication process
5. NETWORK ADMINISTRATION: Introduction to network administration approaches; addressing and subnetting: fixed vs. variable masks, VLAN principles and configuration, routing concepts, static and dynamic routing; routing protocols (RIP, OSPF, BGP)
6. ADVANCED NETWORK MANAGEMENT SERVICES: Configuring a Linux/Windows box as a router; dial-up configuration & authentication: PPP, RAS; configuring a DNS server; configuring Sendmail service; configuring a web server; configuring a proxy server; TCP/IP troubleshooting (ping, traceroute, ifconfig, netstat, ipconfig, network management).
7. NETWORK SECURITY: Security planning; categories of security; access control and monitoring: wrappers; firewalls: filtering rules, detection and prevention of Denial of Service (DOS) attacks; automatic identification of configuration loop holes; security information resources: cert, installing and upgrading system software, use of scripting tools

REFERENCE BOOKS
OBJECTIVE
To provide knowledge about the enterprise resource planning tools, models and techniques

PRE-REQUISITES
Knowledge of internet and web development, data mining, computer networks, software engineering

Unit 1. ENTERPRISE RESOURCE PLANNING – ERP overview; need of ERP; growth of ERP; benefit; Proper and improper ERP implementation; data warehousing; data mining; OLAP.

Unit 2. ERP AND RELATED TECHNOLOGIES: Business process reengineering (BPR); management information system (MIS); decision support systems (DSS); executive support systems (ESS); data warehousing, data mining; online analytical processing (OLTP); supply chain management (SCM); customer relationship management (CRM).

Unit 3. ERP MODULES AND VENDORS: Finance; production planning, control & maintenance; sales & distribution; human resource management (HRM); inventory control system; quality management; ERP market

Unit 4 ERP IMPLEMENTATION LIFE CYCLE: evaluation and selection of ERP package; project planning; implementation team training & testing; end user training & going live; post evaluation & maintenance; introduction to hidden costs, vendors, consultant employees.

Unit 5. ERP & E-COMM, FUTURE DIRECTIVES- in ERP, ERP and internet, critical factors guiding selection and evaluation, strategies for successful implementation, impediments and initiatives to achieve success, critical success and failure factors, integrating ERP into organizational culture.

Unit 6. ERP CASE STUDIES: Post implementation review of ERP packages in manufacturing, services, and other organizations; using ERP tool: either sap or oracle format to case study.

TEXT BOOK

REFERENCE BOOKS
2. Garg, V.K. ,& Venkitakrishnan, N.K. , “ERP Ware: ERP Implementation Framework”
LIST OF EXPERIMENTS
1. Management of the users and the domain.
2. Configuring DHCP.
3. Setting up the local security policy.
4. Start and stop services from user window and command prompt.
5. Use of event viewer.
6. Use of the performance monitor.
7. Management of the IIS and FJP server.
8. Setting up of local area network.
10. Use of utilities: Ping, Tracert, Netstat, Net, IP configuration, Path ping
11. Use of network monitor.
12. Setting up of a DNS.
13. Setting up and use “Terminal Client Services”.

REFERENCE BOOKS

LIST OF EXPERIMENTS
1. Study of Prolog programming language
2. Write programs to use iterative structures using Prolog (at least 3 programs)
3. Write programs to demonstrate inferencing/ deductive logic using Prolog (at least 3 programs)
4. Write a program to solve 8 queens problem using Prolog.
5. Solve any problem using depth first search using Prolog.
7. Solve 8-puzzle problem using best first search using Prolog
10. Write program to exhibit the ability of building an Expert System using Prolog
11. Study the properties and issues of Natural Language Processing
12. Study the grammar mapping issues in language translation from English to Hindi and vice versa

REFERENCE BOOKS
This involves essentially developing software after studying / going through the relevant process/data pertaining to an organization.

The seminar is to cover the details regarding Internship- viz. problem definition, literature survey, concepts and methodology employed, analysis, design and development, conclusions and future work.

**OBJECTIVE**

The Internship course is a formal method of linking university with the world of work and essentially takes the class room for 20-22 weeks to a professional location where the student and faculty solve real-life problems, of course, with the help of professional experts. Resident University faculty will supervise the education of the students.

Internship is conducted at various production and manufacturing units, Design, Development and Consulting Agencies, National Laboratories, R&D Centres, etc. The students solve real-life problems of interest to the host organizations. The professional expert acts as a consultant while resident University faculty supervises the work.
OBJECTIVE
To introduce various technological aspects about parallelism in super computing, microprocessors supporting such high scale computing, other hardware architectures, ultimately leading to high performance computing through grid computing.

1. PARALLEL COMPUTER MODELS: The state of computing; multiprocessors and multicomputers; multivector and SIMD computers; architectural development tracks.

2. PROGRAM AND NETWORK PROPERTIES: Conditions of parallelism; data and resource dependences; hardware and software parallelism; program partitioning and scheduling; grain size and latency; program flow mechanisms; control flow versus data flow; data flow architecture; demand driven mechanisms; comparisons of flow mechanisms.

3. SYSTEM INTERCONNECT ARCHITECTURES: Network properties and routing; static interconnection networks; dynamic interconnection networks; multiprocessor system interconnects; hierarchical bus systems; crossbar switch and multiport memory; multistage and combining network.

4. PROCESSORS AND MEMORY HIERARCHY: Advanced processor technology; instruction-set architectures; CISC scalar processors; RISC scalar processors; superscalar processors; VLIW architectures; vector and symbolic processors; memory technology: hierarchical memory technology, inclusion, coherence and locality, memory capacity planning, virtual memory technology.

5. BACKPLANE BUS SYSTEM: Backplane bus specification; addressing and timing protocols; arbitration transaction and interrupt; cache addressing models; direct mapping and associative caches.

6. PIPELINING: Linear pipeline processor; nonlinear pipeline processor; instruction pipeline design; mechanisms for instruction pipelining; dynamic instruction scheduling; branch handling techniques; arithmetic pipeline design; computer arithmetic principles; static arithmetic pipeline; multifunctional arithmetic pipelines.

7. VECTOR PROCESSING PRINCIPLES: Vector instruction types; vector-access memory schemes; synchronous parallel processing: SIMD architecture and programming principles, SIMD parallel algorithms, SIMD computers and performance enhancement.

REFERENCE BOOKS
OBJECTIVE
To bring out various issues related to advanced computing with respect to database management systems such as parallelism in implementation, data backup and recovery management, intelligent data mining techniques, standards, etc.

PRE-REQUISITES: Knowledge of database management systems

1. DATA MODELS: EER model and relationship to the OO model; object oriented data model and ODMG standard; other data models - NIAM, GOOD, ORM
2. QUERY OPTIMISATION: Query execution algorithms; heuristics in query execution; cost estimation in query execution; semantic query optimisation; database transactions and recovery procedures: transaction processing concepts, transaction and system concepts, desirable properties of a transaction, schedules and recoverability, serializability of schedules; transaction support in SQL; recovery techniques; database backup; concurrency control, locking techniques for concurrency control, concurrency control techniques; granularity of data items
3. CLIENT/SERVER COMPUTING: Client/Server concepts; 2-tier and 3-tier client/server systems; client/server architecture and the internet; client/server models; technology components of client/server systems; application development in client/server systems
4. DISTRIBUTED DATABASES: Reliability and commit protocols; fragmentation and distribution; view integration; distributed database design; distributed algorithms for data management; heterogeneous and federated database systems
5. DEDUCTIVE DATABASES: Recursive queries; Prolog/Datalog notation; basic inference mechanism for logic programs; deductive database systems; deductive object oriented database systems
6. DATA WAREHOUSING: Basic concepts; data warehouse architecture; data characteristics; reconciled data layer data transformations; derived data layer user interface.
7. COMMERCIAL AND RESEARCH PROTOTYPES: Parallel database; multimedia database, mobile database; digital libraries; temporal database

REFERENCE BOOKS
OBJECTIVE
The course will provide a down-to-earth overview of cryptographic techniques applicable in an IT environment, and outline the constraints and limitations of realistic secure systems. A running theme is the tradeoff between usability and security of a system. Also covered are a number of compression techniques - data compression and data encryption are, in some respects, closely related. A working knowledge of C is assumed and essential.

1. **COMPRESSION**: Packing; Huffman coding; run length encoding; Lempel-Ziv-Welch; Phil Katz’s PKZIP; Delta modulation; JPEG.
2. **ERROR DETECTION AND CORRECTION**: Parity; 1, 2, n-dimensions, Hamming codes; p-out-of-q codes
3. **CRYPTOGRAPHY**: Vocabulary; history, steganography – visual, textual; cipher hiding; false errors; public key cryptography - authentication, signatures, deniability
4. **MATHEMATICS**: Information; confusion; diffusion; modular arithmetic; inverses; Fermat’s little theorem, Chinese remainder theorem; factoring; prime numbers; discrete logarithms
5. **ALGORITHMS**: DES; AES (Rijndael); IDEA; one time pad; secret sharing and splitting; RSA; elliptic curves; modes; random numbers
6. **ATTACKING SYSTEMS**: Recognition; destroying data; cryptanalysis - differential cryptanalysis; cracking DES
7. **ENCRYPTION**: Advanced Encryption Standard; Evaluation Criteria for Advanced Encryption Standard; The Advanced Encryption Standard Cipher; Substitute Byte Transformation; Contemporary Symmetric Ciphers; Triple Data Encryption Standard; Blowfish; RC5; Characteristics of Advanced Symmetric Block Ciphers; Confidentiality using Symmetric Encryption; Key Distribution.

**REFERENCE BOOKS**
INTRODUCTION TO XML

CA-1401

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

PRE-REQUISITES: Knowledge of internet and web development, data mining, computer networks, DBMS.

Unit 1 - XML Fundamentals
Introducing the eXtensible Markup Language (XML); Extending and Adopting Markup Languages From SGML to XML and XHTML; Benefits and Drawbacks of XML; Representing Mixed Data and Context with XML; Creating an XML Document; Defining Structure; Rules for Well-Formed and Valid XML; Changing XML Documents

Unit 2 - XML Syntax & Namespaces
Tag Attributes and Naming Rules; Empty and Non-Empty Elements; Processing Instructions for XML; Accessing Data from XML Elements; XML Namespaces; Prefixes and Declarations; Default and Multiple Namespaces

Unit 3 - XML Document Type Definition (DTD)
XML DTD as an XML Schema; Creating a DTD; Element Conditions and Quantifiers; Referencing DTD Declarations; Validating DTD Compliance

Unit 4 - XML Schema Definition (XSD)
Element and Attribute Declarations; Simple, Complex, and Built-in Types; Named and Anonymous Types; Associating XML with a Schema; Validating XSD Compliance

Unit 5 - XQuery and XPath
Why XQuery and XPath; XPath Nodes and Syntax; Seven Node Types; Node Paths and Predicates; Node Axes and Functions; XQuery Structure and Usage; XPath and XSD in XQuery; Terms and Syntax; Selecting and Filtering Elements

Unit 6 - Publishing XML
Stylesheet Languages; Using Style Sheets with XML; Page Layout with Cascading Style Sheets (CSS); CSS Syntax and Classes; Introduction to XSL
Unit 7 - Applying XML
XML and Web Services; HTML with XML; XML and e-Commerce; XML Databases; Storing Binary Data in XML; XML parser.

TEXT BOOK

REFERENCE Books
OBJECTIVE To develop deep understanding about computer software testing methodologies and tools

PRE-REQUISITES Knowledge of programming, software engineering,

1. FUNDAMENTALS AND TESTING TYPES: First, second and later cycles of testing, Objectives and limits of testing, Overview of software development stages, Planning and Design stages and testing during these stages. Glass box code, Regression and Black box testing, Software errors, Categories of software error

2. REPORTING AND ANALYZING BUGS: Problem reports, Content and Characteristics of Problem Report, analysis and Tactics for analyzing a reproducible bug, Making a bug reproducible

3. PROBLEM TRACKING SYSTEM: Objective of Problem Tracking System, tasks of the system, Problem tracking overview, users of the tracking system, mechanics of the database

4. TEST CASE DESIGN: Characteristics of a good test, equivalence classes and boundary values, visible state transitions, Race conditions and other time dependencies, load testing. Error guessing, Function equivalence testing, Regression Testing, General issues in configuration testing, printer testing

5. LOCALIZATION AND USER MANUALS TESTING: Translated text expands, Character sets, Keyboards, Text filters, Loading, saving, importing, and exporting high and low ASCII, Operating system Language, Hot keys, Error message identifiers, Hyphenation rules, Spelling rules, Sorting Rules, Uppercase and Lowercase conversion, Printers, Sizes of paper, CPU's and video, Rodents, Data formats and setup options, Rulers and measurements, Culture-bound Graphics and output, European product compatibility, Memory availability, automated testing, Testing User Manuals, Effective documentation, documentation tester's objective, How testing documentation contributes to software reliability

6. TESTING TOOLS AND TEST PLANNING: Fundamental tools, Automated acceptance and regression tests, standards, Translucent box testing Overall objective of the test plan: product or tool? Detailed objective, type of test, strategy for developing components of test planning documents, components of test planning documents, documenting test materials

7. MANAGEMENT ISSUES OF TESTING: Software Development tradeoffs and models, Quality-related costs, The development time line, Product design, alpha, Pre-beta, Beta, User Interface freeze, Pre-final, Final integrity testing, Project post-mortems, Legal consequences of defective software, Managing and role of a testing group, independent test agencies
MCA

TEXT BOOK

REFERENCE BOOKS
OBJECTIVE
The goal is to relay the theoretical and practical fundamental knowledge of neural networks and studying its analogy to human brain.

PRE-REQUISITES
Knowledge of mathematics, computer architecture and organization

1. **OVERVIEW OF BIOLOGICAL NEURONS:** Structure of biological neurons relevant to ANNs.
2. **FUNDAMENTAL CONCEPTS OF ARTIFICIAL NEURAL NETWORKS:** Models of ANNs; Feed forward and feedback networks; learning rules: Hebbian learning rule, perception learning rule, delta learning rule, Widrow-Hoff learning rule, correction learning rule, Winner-take-all learning rule, etc.
3. **SINGLE LAYER PERCEPTION CLASSIFIER:** Classification model, features and decision regions; training and classification using discrete perceptron, algorithm, single layer continuous perceptron networks for linearly separable classifications.
4. **MULTI-LAYER FEED FORWARD NETWORKS:** linearly non-separable pattern classification; delta learning rule for multi-perceptron layer; generalized delta learning rule, error back-propagation training; learning factors; examples.
5. **SINGLE LAYER FEED BACK NETWORKS:** Basic concepts; Hopfield networks; training and examples.
6. **ASSOCIATIVE MEMORIES:** Linear association, basic concepts of recurrent auto associative memory: retrieval algorithm, storage algorithm; bi-directional associative memory, architecture, association encoding and decoding, stability.
7. **SELF ORGANIZING NETWORKS:** Unsupervised learning of clusters, winner-take-all learning, recall mode, Initialization of weights, separability limitations

**TEXT BOOK**

**REFERENCE BOOKS**
OBJECTIVE
The main objective behind this course is to learn about the various network attacks and preventing attacks. This course is designed to cover Application security, Operating system security, Network security, Web security etc.

PRE-REQUISITES
Knowledge of data communications and computer networks, computer programming, data structures, mathematics, telecom network. Knowledge of digital signal processing is desirable

1. INTRODUCTION: Codes and ciphers; some classical systems; statistical theory of cipher systems: complexity theory of crypto systems; stream ciphers, block ciphers.
2. STREAM CIPHERS: Rotor based system; shift register based systems; design considerations for stream ciphers, crypt-analysis of stream ciphers; combined encryption and encoding; block ciphers: DES and variant, modes of use of DES; public key systems: knapsack systems, RSK, Diffie Hellman exchange; authentication and digital signatures; elliptic curve based systems.
3. SYSTEM IDENTIFICATION AND CLUSTERING: Cryptology of speech signals: narrow band and wide band systems; analogue and digital Systems of speech encryption.
4. SECURITY: HASH FUNCTION – AUTHENTICATION: Protocols; digital signature standards; electronic mail security: PGP (Pretty Good Privacy), MIME; data compression technique; IP security: architecture, authentication leader, encapsulating security; payload: key management; web security: secure socket layer & transport layer security, secure electronics transactions; firewalls design principle; established systems.
5. TELECOMMUNICATION NETWORK ARCHITECTURE: TMN management layers, management information model; management servicing and functions; structure of management information and TMN information model; SNMP v1, SNMP2 & SNMP3, RMON1 & 2; Broadband Network Management (ATM, HFC, DSL); ASN
7. LEGAL, PRIVACY, AND ETHICAL ISSUES IN COMPUTER SECURITY: Protecting program and data; information and law; rights of employees and employers; software failures; computer crime, privacy; ethical issues in computer society; case studies of ethics

REFERENCE BOOKS

Unit 1 – Basics of Animation
Introduction to animation; 12 basic principles of animation; Inbetweening; Using Key Points; Superimposition; Arcs; Head Turns; Eye Movement; Walking; Running; Timing; Anticipation; Realistic Touches; Exaggerated Action; Special Effects.

Unit 2 – Basic Operations & 3D Modelling
Moving, Rotating and scaling the objects; Edit a Mesh; Sculpting Modelling.

Unit 3 – UV Mapping, Curves and NURBS
Creating a UV Map; Texture Painting; Projection Painting; Metaballs; Curves; Spin and NURBS.

Unit 4 - Rigging and Animation
Key-framing with the Timeline; Pivot Point: The Center of Rotation; Basic Tracking: Eyes That Follow; Rigging with Bones; Rigging a Simple Character

Unit 5 - Making Movies
The Compositing Node Editor; Lighting Adjustments; A Practical Example of Compositing; The Video Sequence Editor; Making Particles; Making Hair; Fluid Dynamics; Smoke; Soft Body Physics.

Unit 6 - Gaming
Creation of a simple game world and a fully rigged character for a game; controlling characters and scenes with logic blocks Game Engine Physics; Creating Your Own Droid; Silly Soccer Game; A Change of Scene.
Unit 7 – Game scripting with Python
Introduction to python; installation and configuration of python; writing programs with python; Python to create more streamlined, organized, and powerful game logic.

REFERENCE Books
3. Animating with blender, D. Orland Hess; Focal press.
4. Beginning Blender, Lance Flavell; Apress.

Web Site REFERENCE
1. www.Blenderguru.com
2. www.Blender.org
3. www.python.org
OBJECTIVE
To lay adequate foundation for design and development of compiler and other system software tools such as linkers, debuggers, assemblers, etc.

PRE-REQUISITES
Knowledge of data structures, basic programming concepts, theory of computations and operating systems

1. INTRODUCTION: Evolution of Components Systems Programming, Assemblers, Loaders, Linkers, Macros, Compilers, Software tools: Text editors, Interpreters, program generators, Testing software, Programming environment (such as Integrated Development Editors)

2. SYSTEM SOFTWARE SPECIFICS: Compiler: Brief overview of compilation process, Incremental compiler, structure of compiler: its different phases, Compiler construction tools. Assembler: Problem statement, single phase and two phase assembler, symbol table; Loader schemes, compile and go Loader, general loader schemes, absolute loader, Subroutine linkage, Reallocating loader, Direct linkage Loader, Binders, Linking loader, overlays

3. LEXICAL AND SYNTAX ANALYSIS: Role of lexical analyzer, design of lexical analyzer, regular expressions, Specification and recognition of tokens, input buffering, a language specifying lexical analyzer. Finite automata, conversion from regular expression to finite automata, and vice versa, minimizing number of states of DFA, Implementation of lexical analyzer. Syntax Analysis: Role of parsers, context free grammars, definition of parsing

4. PARSING TECHNIQUE: Shift- reduce parsing, operator precedence parsing, top down parsing, predictive parsing, LR parsers, SLR, LALR and Canonical LR parser

5. SYNTAX DIRECTED TRANSLATIONS: Syntax directed definition, construction of syntax trees, syntax directed translation scheme, implementation of syntax directed translation, three address code, quadruples and triples

6. SYMBOL TABLE & ERROR DETECTION AND RECOVERY: Symbol tables, its contents and data structure for symbol tables; trees, arrays, linked lists, hash tables. Errors, lexical phase error, syntactic phase error, semantic error

7. CODE OPTIMIZATION AND CODE GENERATION: Code generation, forms of objects code, machine dependent code, optimization, register allocation for temporary and user defined variables

REFERENCE BOOKS
OBJECTIVE
To introduce about incorporating more mathematical approach (beyond conventional logic system) into the artificial intelligence approaches for problem solving such as fuzzy logic, genetic algorithms, etc.

1. **INTRODUCTION:** Comparison of soft computing methods: neural networks, fuzzy logic, genetic algorithm with conventional artificial intelligence (hard computing).
2. **OPTIMIZATION:** Least-square methods for system identification, recursive least square estimator; LSE for nonlinear models; derivative based optimization: descent methods, Newton’s method, conjugate gradient methods; nonlinear least-squares problems: Gauss Newton method, Levenberg- Marquardt method.
3. **NEURAL NETWORKS:** Different architectures; back-propagation algorithm; hybrid learning rule; supervised learning: perceptrons, back-propagation multilayer perceptrons, radial basis function networks; unsupervised learning – competitive learning network, Kohonen self-organizing networks, the Hopfield network.
4. **FUZZY SET THEORY:** Basic definition and terminology; basic concepts of fuzzy logic; set theoretic operators; membership functions: formulation and parameterization; fuzzy union, intersection and complement; fuzzy rules and fuzzy reasoning; fuzzy inference systems: Mamdani and Sugeno fuzzy models.
5. **NEURO-FUZZY MODELLING:** Adaptive neuro-fuzzy inference systems; neuro-fuzzy controller-feedback control; back propagation through time and real-time recurrent learning; gradient-free optimization.
6. **NEURO-FUZZY CONTROLLER IN ENGINEERING APPLICATIONS:** Analytical issues in fuzzy logic control; fuzzy logic in intelligent agents.
7. **GENETIC ALGORITHMS:** Basics of genetic algorithms; design issues in genetic algorithm; genetic modeling; sequential and parallel genetic algorithms.

REFERENCE BOOKS
5. Sivanandam, “Introduction to Neural Networks with MATLAB 6.0”, Tata McGraw Hill
OBJECTIVE
There is a long-standing interest from an increasing number of disciplines in migrating interactive computer graphics away from traditional keyboard/monitor/mouse interaction and out into the broader environment. Applications commonly obtain data from disparate media sources (e.g., sound, video, network data feeds, sensors, etc.); process this data in real-time, and ultimately (re)present information in different forms. Movement becomes color, video controls sound, and light drives motion.

1 INTRODUCTION TO WIRELESS TECHNOLOGIES: WAP services; serial and parallel communication; asynchronous and synchronous communication; FDM; TDM; TFM; spread spectrum technology.
2 INTRODUCTION TO BLUETOOTH: Specification; core protocols; cable replacement protocol Bluetooth radio; type of antenna; antenna parameters; frequency hoping.
3 BLUETOOTH NETWORKING: Wireless networking; wireless network types; devices roles and states; Ad-hoc Network; scatternet
4 CONNECTION ESTABLISHMENT PROCEDURE: Notable aspects of connection establishment; mode of connection; Bluetooth security; security architecture; security level of services; profile and usage model: generic access profile (gap), SDA, serial port profile; secondary Bluetooth profile.
5 HARDWARE: Bluetooth implementation; baseband overview; packet format; transmission buffers; protocol implementation: link manager protocol; logical link control adaptation protocol; host control interface; protocol interaction with layers
6 PROGRAMMING WITH JAVA: Java programming; J2ME architecture; Javax; Bluetooth package interface; classes; exceptions, Javax. Obex package: interfaces, classes
7 BLUETOOTH SERVICES REGISTRATION AND SEARCH APPLICATION; Bluetooth Client and Server Application. Overview of IRDA, homeRF, Wireless LANS, JINI

REFERENCE BOOKS
OBJECTIVE
This course will introduce the algorithms and technologies of distributed systems. It will teach both fundamentals as well as systems where these fundamentals are applied in practice.

PREREQUISITES
Knowledge of databases, networking, operating system and web technologies

1. DISTRIBUTED COMPUTING: History, forms of computing; strengths and weaknesses of distributed computing; OS basics; network basics; software engineering basics; CLIENT SERVER PARADIGM: issues, software engineering for a network service, connection oriented and connectionless servers, iterative server and concurrent server, stateful servers.

2. INTERPROCESS COMMUNICATION: Archetypal IPC program interface; event synchronization; timeouts and threading; deadlock and timeouts; data representation, data encoding; text based protocols, request response protocols; event and sequence diagram; connection vs. connectionless IPC.

3. DISTRIBUTED COMPUTING PARADIGMS AND SOCKET API: Paradigms; abstraction; socket metaphor; diagram socket API, stream mode socket API; sockets with non-blocking I/O; secure socket API.

4. GROUP COMMUNICATION: Unicasting; multicasting, archetypal multicast API; connection oriented and connectionless; reliable, unreliable multicast; Java basic multicast API.

5. DISTRIBUTED OBJECTS: Message passing vs. distributed objects; archetypal distributed object architecture; distributed object systems; remote procedure calls; Java RMI architecture; API for Java RMI; Advanced RMI: Client callback, stub downloading, RMI security manager; allowing for stub downloading.

6. SIMPLE OBJECT ACCESS PROTOCOL: SOAP request, SOAP response; Apache SOAP; invoking web service; implementing web service.

7. ADVANCED DISTRIBUTED COMPUTING PARADIGMS: Message queue system paradigm; mobile agents; network service; object spaces.

REFERENCE BOOKS
OBJECTIVE
Using a “building block” approach, the ISM curriculum provides a core understanding of storage technologies and progresses into system architectures, introduction to networked storage, and introduction to information availability. The course provides a comprehensive introduction to data storage technology fundamentals. Students will gain knowledge of the core logical and physical components that make up a storage systems infrastructure.

PRE-REQUISITES
Knowledge of computer networks

1. INTRODUCTION: Meeting today’s data storage needs - data creation; data creation: individuals, business; categories of data; data storage models; common data storage media and solutions - tape storage systems, optical data storage, disk based storage
2. DATA CENTER INFRASTRUCTURE: Example; key requirements of storage systems management activities
3. STORAGE SYSTEMS ARCHITECTURE: Storage system environment; components of a host; connectivity; physical disks; RAID array; disk storage systems; data flow exercise
4. NETWORKED STORAGE: Direct Attached Storage (DAS), Network Attached Storage (NAS), Fiber Channel Storage Area Network (FC SAN), IP Storage Area Network (IP SAN), Content Addressed Storage (CAS)
5. BUSINESS CONTINUITY: Introduction, overview, backup and recovery, local replication, remote replication.
6. MONITORING AND MANAGING THE DATA CENTER: Areas of the data center to monitor; considerations for monitoring the data center; techniques for managing the data center.
7. SECURING STORAGE AND STORAGE VIRTUALIZATION: Securing the storage infrastructure; virtualization technologies.

REFERENCE BOOKS
OBJECTIVE:

Upon completion of this course the student will Know the basic concepts and technique of developing applications for the Android phone, Be able to use the SDK and other development tools, Know the basic concepts of Android phone features and capabilities.

PRE-REQUISITES: previous programming experience in Java

Unit 1: Getting Started with Android Programming

Unit 2: Activities and Intents
Understanding Activities, Applying Styles and Themes to Activity, Hiding the Activity Title, Displaying a Dialog Window, Displaying a Progress Dialog, Linking Activities Using intents, Resolving Intent Filter Collision, Returning Results from an Intent, Passing Data Using an Intent Object, Calling Built-in Applications Using intents, Understanding the Intent Object, Using Intent Filters, Adding Categories, Displaying notifications

Unit 3: Getting to know the Android user interface
Understanding the Components of a Screen, Views and ViewGroup, LinearLayout, AbsoluteLayout, TableLayout, RelativeLayout, FrameLayout, ScrollView, Adapting to Display Orientation, Anchoring Views, Resizing and Repositioning, Managing Changes to Screen Orientation, Persisting State Information during Changes in Configuration, Detecting Orientation Changes, Controlling the Orientation of the Activity, Creating the User interface Programmatically, Listening for UI notifications, Overriding Methods Defined in an Activity, Registering Events for Views, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioGroup Views, Progress bar View, AutoCompleteTextView

74
Unit 4: Storing and Retrieving Data
Saving and Loading User Preferences, Persisting Data to Files, using the file system, Saving to Internal Storage, Saving to External Storage (SD Card), Choosing the Best Storage Option Using Static Resource, Creating and Using Databases, Creating the DBAdapter Helper Class, Using the Database Programmatically, Upgrading the Database, Pre-Creating the Database, Bundling the Database with an Application, working with content provider classes

Chapter 5: Messaging and networking
SMS Messaging, Sending SMS Messages Programmatically, Getting Feedback After Sending the Message, Sending SMS Messages Using Intent, Receiving SMS Messages, Upgrading an Activity from a BroadcastReceiver, Invoking an Activity from a , BroadcastReceiver, Caveats and Warnings, Sending e-Mail, Networking, Downloading Binary Data, Downloading Text Files, Accessing Web Services, Performing Asynchronous Calls

Unit-6: Graphics, Animation and Multimedia

Unit 7: Developing Android Services & Publishing Android Applications
Creating Your Own Services, Performing Long-Running Tasks in a Service, Performing Repeated Tasks in a Service, Executing Asynchronous Tasks on Separate Threads Using IntentService, Communicating between a Service and an Activity, Binding Activities to Services, Preparing for Publishing, Digitally Signing Your Android Applications, Deploying APK Files, Using the adb exe Tool, Using a Web Server, Publishing on the Android Market, Creating a Developer Profile, Submitting Your Apps

Books:
OBJECTIVE
Embedded computers are found everywhere from home appliances to automobiles to medical devices. Designing an embedded computing system is a challenging task because the requirements include manufacturing cost, performance, power consumption, user interface, hard deadlines and rich functionality. The objective is to teach embedded system design process which includes requirements, specification, architecture, components and system integration phases. The course will have real-life design examples to illustrate the design process and the students are encouraged to design embedded systems to gain experience.

PRE-REQUISITES
Knowledge of logic design, assembly language programming, computer organization and architecture, microprocessors and interfacing, operating systems

1. INTRODUCTION: Introduction, overview, design process; instruction set architecture; CISC and RISC instruction set; architecture; basic embedded processor/ microcontroller architecture; memory system architecture; I/O sub-system; co-processors and hardware accelerators; processor performance enhancement; 16 & 32 bit microprocessor and microcontroller and DSP hardware with reference to embedded system.

2. REAL TIME OPERATING SYSTEMS: Real time operating system overview; basic features of an operating system, kernel features; processes and threads, context switching; scheduling, inter-process communication; real-time memory management; I/O processes; exposure to Windows CE, QNX, micro kernels and c/OS of introduction to process models; interrupt routines in an RTOs environment; encapsulating semaphores and queues; hard real-time scheduling considerations; saving memory space.

3. DESIGNING EMBEDDED COMPUTING PLATFORM: Using CPU bus, memory devices and their characteristics, I/O devices, component interfacing, memory interfacing; I/O device interfacing, interfacing protocols, designing with processors: system architecture, hardware design, FPGA based design; implementation: development environment, debugging techniques, design examples: data compressor, alarm clock.

4. PROGRAMMING EMBEDDED SYSTEMS: Program design, programming languages, use of high level languages, programming and run-time environment, basic compilation techniques, analysis and optimization of execution time, analysis and optimization of energy and power, analysis and optimization of program size, program validation and testing

5. NETWORK BASED EMBEDDED APPLICATIONS: Network fundamentals, layers and protocols, network architectures, distributed embedded architectures, elements of protocol design, high level protocol
design languages, network based design, internet-enabled systems: protocols for industrial and control applications; internetworking protocols; wireless applications

6. **EMBEDDED CONTROL APPLICATIONS:** Introduction, open-loop and closed loop control systems; PID controllers, fuzzy logic controller; application examples: washing machine, automotive systems, auto-focusing digital camera, air-conditioner

7. **EMBEDDED SYSTEM DEVELOPMENT:** Design methodologies; architectural design; design examples: telephone PBX, PDA, set-top box, elevator control system, ATM system, fault-tolerance techniques, reliability evaluation techniques

**TEXT BOOK**
Simon David E., —An Embedded System Primerl, Addison-Wesley, 1999

**REFERENCE BOOKS**
OBJECTIVE
Recent developments in portable devices and high-bandwidth, ubiquitous wireless networks has made mobile computing a reality. Indeed, it is widely predicted that within the next few years’ access to Internet services will be primarily from wireless devices, with desktop browsing the exception. Such predictions are based on the huge growth in the wireless phone market and the success of wireless data services. This course will help in understanding fundamental concepts, current developments in mobile communication systems and wireless computer networks.

PRE-REQUISITES
Computer Networks and wireless communication

1. **INTRODUCTION TO WIRELESS TRANSMISSION:** Applications, A short history of wireless communication, Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems.
2. **MEDIUM ACCESS CONTROL:** Motivation for a specialized MAC: Hidden and Exposed terminals, Near and Far terminals; SDMA, FDMA, TDMA: Fixed TDM, Classical Aloha, Slotted Aloha, Carrier sense multiple access, Demand assigned multiple access, PRMA packet reservation multiple access, Reservation TDMA, Multiple access with collision avoidance, Polling, Inhibit sense multiple access; CDMA: Spread Aloha multiple access
3. **TELECOMMUNICATION SYSTEMS:** GSM: Mobile services, System architecture, Radio interface, Protocols, Localization And Calling, Handover, Security, New data services; DECT: System architecture, Protocol architecture; TETRA, UMTS and IMT-2000: UMTS Basic architecture, UTRA FDD mode, UTRA TDD mode
4. **SATELLITE & BROADCAST SYSTEMS:** History, Applications, Basics: GEO, LEO, MEO; Routing, Localization, Handover, Examples, Cyclic repetition of data, Digital audio, broadcasting: Multimedia object transfer protocol; Digital video broadcasting
6. **MOBILE NETWORK LAYER:** Mobile IP: Goals, assumptions and requirements, Entities and terminology, IP packet delivery, Agent advertisement and discovery, Registration, Tunnelling and Encapsulation , Optimizations, Reverse tunnelling, Ipv6; Dynamic host configuration protocol, Ad hoc networks: Routing, Destination sequence distance vector, Dynamic source routing, Hierarchical algorithms, Alternative metrics,
7. **MOBILE TRANSPORT LAYER:** Traditional TCP: Congestion control, Slow start, Fast retransmit/fast recovery, Implications on mobility; Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/fast recovery, Transmission/time-out freezing, Selective retransmission, Transaction oriented TCP

REFERENCE BOOKS
1. Jochen Schiller, “Mobile Communications”, Addison Wesley/Pearson Education, 2005

OBJECTIVE
To provide adequate knowledge about the different types of system software available & to introduce the object oriented concepts to the programming skills.

PRE-REQUISITES
Understanding of object orientation and knowledge of software engineering

1. REVIEW OF OBJECT ORIENTED SYSTEMS: Design objects, class hierarchy, inheritance, polymorphism, object relationships and associations, aggregations and object containment, object persistence, meta classes, object oriented systems development life cycle, Software development process, object oriented systems development: a use case driven approach.

2. OBJECT ORIENTED ANALYSIS: Analysis process, use case driven object oriented analysis, use-case model, object classification, theory, different approaches for identifying classes, classes, responsibilities and collaborators, identifying object relationships, attributes and methods, super sub class relationships, A-part of relationships aggregation, class responsibilities, object responsibilities.

3. OBJECT ORIENTED DESIGN: Object oriented design process, corollaries, design axioms, design patterns, object oriented design philosophy.

4. METHODOLOGY FOR OBJECT ORIENTED DESIGN: Object modeling technique as software engineering methodology, Rumbaugh methodology, Jacobson Methodology, Booch Methodology.

5. UNIFIED APPROACH FOR OBJECT ORIENTED DESIGN: Patterns, Frameworks, the unified approach, unified modeling language (UML).

6. UML: Why we model, types of models, principles of modelling, object oriented modelling, object oriented concepts, UML notation, object oriented analysis: use case diagrams, interaction diagrams, activity diagrams, object oriented design: class diagrams, object diagrams, state diagrams, collaboration diagrams, post-testing: deployment diagrams, patterns, frameworks.

7. USING UML FOR OOD: UML object constraint language, designing classes: the process, class visibility, refining attributes, designing methods and protocols, packages and managing classes, designing interface objects, view layer interface design, macro and micro level interface design process.

REFERENCE BOOKS
OBJECTIVE
To highlight the issues in knowledge representation, learning and understanding by a computer obtained from real world, and to take decisions based on them

1. ARTIFICIAL INTELLIGENCE: History and applications; AI problems and techniques; concept of AI; approaches: acting and thinking like humans and rationally; brief history of AI; foundations of AI; underlying assumptions; application areas.

2. PRODUCTION SYSTEMS, STRUCTURES AND STRATEGIES FOR STATE SPACE SEARCH: Data driven and goal driven search; depth first and breadth first search; DFS with iterative deepening.

3. HEURISTIC SEARCH: Best first search; A* algorithm; AO* algorithm; constraint satisfaction; using heuristics in games: Minimax search, alpha beta procedure; state space theory/representation.

4. KNOWLEDGE REPRESENTATION: Simple relational knowledge; inheritable knowledge; inferential knowledge; procedural knowledge propositional calculus; predicate calculus; theorem proving by resolution; answer extraction; AI representational schemes: semantic nets, conceptual dependency, scripts, frames; introduction to agent based problem solving.

5. MACHINE LEARNING: Symbol based and connectionist; social and emergent models of learning; the genetic algorithm: genetic programming; overview of expert system technology: rule based expert systems; introduction to natural language processing; neural networks.

6. LANGUAGES AND PROGRAMMING TECHNIQUES FOR AI: Introduction to PROLOG and LISP; search strategies and logic programming in LISP; production system examples in PROLOG.

7. KNOWLEDGE BASED SYSTEMS: Expert systems; components; characteristic features of expert systems; applications; rule based system architecture; representing and using domain knowledge; expert system shell; explaining the reasoning and knowledge acquisition; applications.

REFERENCE BOOKS
OBJECTIVE
To motivate understanding of issues related to natural language understanding, generation and translation, which ultimately linked to machine learning, computer vision and expert systems. This course provides an introduction to the field of computational linguistics, also called natural language processing (NLP) - the creation of computer programs that can understand and generate natural languages (such as English). Natural language understanding as a vehicle will be used to introduce the three major subfields of NLP: syntax (which concerns itself with determining the structure of an utterance), semantics (which concerns itself with determining the explicit truth-functional meaning of a single utterance), and pragmatics (which concerns itself with deriving the context-dependent meaning of an utterance when it is used in a specific discourse context). The course will introduce both knowledge-based and statistical approaches to NLP, illustrate the use of NLP techniques and tools in a variety of application areas, and provide insight into many open research problems.

PRE-REQUISITES
Knowledge of theory of computations

1. INTRODUCTION TO NATURAL LANGUAGE UNDERSTANDING: The study of language; applications of NLP; evaluating language understanding systems; different levels of language analysis; representations and understanding; organization of natural language understanding systems; linguistic background: an outline of English syntax.

2. GRAMMARS AND PARSING: Grammars and sentence structure; top-down and bottom-up parsers; transition network grammars; top-down chart parsing; feature systems and augmented grammars: basic feature system for English

3. MORPHOLOGICAL ANALYSIS AND THE LEXICON: Brief review of regular expressions and automata; finite state transducers; parsing with features; augmented transition networks

4. GRAMMARS FOR NATURAL LANGUAGE: Auxiliary verbs and verb phrases; movement phenomenon in language; handling questions in context-free grammars; hold mechanisms in ATNs.

5. HUMAN PREFERENCES IN PARSING: Encoding uncertainty; deterministic parser; word level morphology and computational phonology; basic text to speech; introduction to HMMs and speech recognition, parsing with CFGs; probabilistic parsing; representation of meaning.

6. AMBIGUITY RESOLUTION: Statistical methods; estimating probabilities; part-of-speech tagging; obtaining lexical probabilities; probabilistic context-free grammars; best first parsing.

7. SEMANTICS AND LOGICAL FORM: Word senses and ambiguity, encoding ambiguity in logical form, semantic analysis; lexical semantics; word sense; disambiguation; discourse understanding; natural language generation, Indian language case studies.

REFERENCE BOOKS
OBJECTIVE
To introduce the students about the basic concepts and analytical methods of processing digital signals, especially, the images and imaging part; to understand the properties of static and streaming images/video.

PRE-REQUISITES
Knowledge of data compression, discrete structures, digital signal processing, computer graphics

1. INTRODUCTION AND DIGITAL IMAGE FUNDAMENTALS: Origins of digital image processing; examples of fields that use digital image processing; fundamentals steps in image processing; elements of digital image processing systems; image sampling and quantization; some basic relationships like neighbours; connectivity, distance measures between pixels; linear and non linear operations.

2. IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN: Some basic gray level transformations; histogram processing; enhancement using arithmetic and logic operations; basics of spatial filters, smoothening and sharpening spatial filters, combining spatial enhancement

3. IMAGE ENHANCEMENT IN THE FREQUENCY DOMAIN: Introduction to Fourier transform and the frequency domain, smoothing and sharpening frequency domain filters; homomorphic filtering; image restoration: a model of the image degradation / restoration process, noise models, restoration in the presence of noise only spatial filtering, periodic noise reduction by frequency domain filtering; linear position-invariant degradations; estimation of degradation function; inverse filtering; Wiener filtering, constrained least square filtering, geometric mean filter; geometric transformations.

4. IMAGE COMPRESSION: Coding; inter-pixel and psycho visual redundancy; image compression models; elements of information theory; error free compression; lossy compression; image compression standards.

5. IMAGE SEGMENTATION: Detection of discontinuities; edge linking and boundary detection; thresholding; region oriented segmentation; motion based segmentation

6. REPRESENTATION AND DESCRIPTION: Representation, Boundary Descriptors, Regional Descriptors, Use of Principal Components for Description, Introduction to Morphology, Some basic Morphological Algorithms.

7. OBJECT RECOGNITION: Patterns and Pattern Classes, Decision-Theoretic Methods, Structural Methods.

REFERENCE BOOKS
MCA