

Computer Science and Engineering Curriculum

E1 Sem1 Courses and Syllabus		
Course Code: CS1102	Course Name: Introduction to Computer Science-I	No. of Credits: 4
Detailed Syllabus: Introduction, C Tokens: Character set, Variables, Keywords, Data types and sizes, Type qualifiers, Numeric Constants and their forms of representation, Character Constants, String Constants, Declarations and Initialization of variables. Operators & Expressions: Conditional Statements: Blocks, If-Else statement, Else-If statement and Switch statement. Iterative Statements: While loop, For loop, Do-While loop, Break, and continue. Arrays: One - dimensional and character arrays, Two-dimensional numeric arrays. Functions: Function Definition, types of User Defined Functions, Parameter passing mechanisms, and simple recursion. Scope & extent: Scope rules, Storage Classes, Multi-file compilation. Pointers: Pointers Arithmetic, Character array of pointers, Dynamic memory allocation, array of Pointer, Pointer to arrays. Structures: Structures, Array of structures, structures within structures, Pointer to structures, self referential structures, Unions. Files: File Handling functions, File error handling functions, Command-line arguments.		
Course Code: MA2101	Course Name: Discrete Mathematics	No. of Credits: 4
Detailed Syllabus: Propositional logic: Syntax, semantics, valid, satisfiable and unsatisfiable formulas, encoding and examining the validity of some logical arguments. Proof techniques: forward proof, proof by contradiction, contra positive proofs, proof of necessity and sufficiency. Sets, relations and functions: Operations on sets, relations and functions, binary relations, partial ordering relations, equivalence relations, principles of mathematical induction, Finite and infinite sets, countable and uncountable sets, Cantor's diagonal argument and the power set theorem. Introduction to counting: Basic counting techniques, inclusion and exclusion, pigeon-hole principle, permutation, combination, summations. Introduction to recurrence relation and generating function. Introduction to graphs: Graphs and their basic properties, degree, path, cycle, subgraphs, isomorphism, Eulerian and Hamiltonian walks, graph coloring, planar graphs, trees. Algebraic structures and morphisms: Algebraic structures with one binary operation and semi groups, monoids and groups, congruence relation and quotient structures. Free and cyclic monoids and groups, permutation groups, substructures, normal subgroups. Algebraic structures with two binary operations and rings, integral domains and fields. Boolean algebra and Boolean ring.		
Course Code: CS1103	Course Name: IT Workshop	No. of Credits: 4
Detailed Syllabus: 1. Identification of components of Motherboard. Installation of MS Windows		

and ubuntu on systems.

2 Introduction to Sci-lab: - VectorOperations, MatrixOperations, Scripts and Functions, ConditionalBranching, Iteration, Plotting 2D graphics, 10Xcos, File handling, Userdefined input and output, Integration, Solving Nonlinear equations, Linear equations, Linear equations using iterative Methods, Interpolation, ODE Euler method, ODE Application, Optimization using karmarkar, Control systems, Discrete systems, Calling user Defined Functions in Xcos.

3. Introduction to latex: -Resume, LATEX typesetting basics, LATEX math typesetting, tables and matrices, graphics, packages, user definable, document class, beamer.

4. Introduction to Beamer: -presentation of Resume using beamer, Write a Latex code adding graphics/image in Beamer

Course Code:	Course Name: Network Analysis	No. of Credits: 4
---------------------	--------------------------------------	--------------------------

Detailed Syllabus:

Course Code:	Course Name: Engineering Mathematics	No. of Credits: 4
---------------------	---	--------------------------

Detailed Syllabus: UNIT-1: SEQUENCES AND SERIES:

Sequences – series – Convergences and divergence – Ratio test – Comparison test – Integral test – cauchy's root test – Raabe's test – Absolute and conditional convergence.

UNIT-2: Differential Calculus (Functions of several variables):

Limit, Continuity and Differentiability of functions of several variables, Partial derivatives and their geometrical interpretation, Differentials, Derivatives of Composite and Implicit functions, Derivatives of higher order and their commutativity, Euler's theorem on homogeneous functions, Harmonic functions, Taylor's expansion of functions of several variables, Maxima and Minima of functions of several variables - Lagrange's method of multipliers.

UNIT-3: Integral Calculus:

Convergence of improper integrals, tests of convergence, Beta and Gamma functions - elementary properties, differentiation under integral sign, differentiation of integrals with variable limits - Leibnitz rule. Rectification, double and triple integrals, computations of

surface and volumes, change of variables in double integrals.

UNIT-4: Vector Calculus:

Scalar and vector fields, level surfaces, directional derivative, Gradient, Curl, Divergence, Laplacian, line and surface integrals, theorems of Green, Gauss and Stokes, orthogonal curvilinear coordinates.

Course Code: CS1702

**Course Name: Introduction to Computer
Science-I Lab**

No. of Credits: 2

Detailed Syllabus:

1. A program for electricity bill taking different categories of users, different slabs in each category. (Using nested if else statement).

2. Write a C program to evaluate the following(using loops):

- a. $1 + x^2/2! + x^4/4! + \dots$ upto ten terms
- b. $x + x/3! + x/5! + \dots$ upto 7 digit accuracy
- c. $1 + x + x^2/2! + x^3/3! + \dots$ upto n terms
- d. Sum of $1 + 2 + 3 + \dots + n$

3. A menu driven program to check the number is:

- i) Prime or not
- ii) Perfect or Abundant or deficient
- iii) Armstrong or not
- iv) Strong or not

v) Fibonacci or not

4. A menu driven program with options (using one -Dimensional array)

- (i) To insert an element into array
- (ii) To delete an element
- (iii) To print elements
- (iv) To print elements in reverse order
- (v) To remove duplicates

5. A menu driven program to display statistical parameters (using one –dimensional array)

- i) Mean ii) Mode iii) Median iv) Variance v) Standard deviation

6. A menu driven program with options (using two dimensional array)

- (i) To compute $A+B$
- (ii) To compute $A \times B$

- (iii) To find transpose of matrix A
- (iv) To Check $A=B$, Where A and B are matrices. Conditions related to size to be tested

7. A menu driven program with options (using Two-dimensional Character arrays)

- (i) To insert a student name
- (ii) To delete a name
- (iii) To sort names in alphabetical order
- (iv) To print list of names
- (v) To print names having maximum length, min. length

8. A menu driven program with options (using Character array of pointers)

- (i) To insert a student name
- (ii) To delete a name
- (iii) To sort names in alphabetical order
- (iv) To print list of names
- (v) To print names having maximum length, min. length

9. Write a program to perform the following operations on Rational numbers (using Structures & pointers):

- i) Read a Rational number
- ii) Addition of two Rational numbers
- iii) Subtraction of two Rational numbers
- iv) Multiplication of two Rational numbers
- v) Division of two Rational numbers
- vi) Display a Rational number

10. Create a student data file (roll no., name, date of birth, rank) and code a program with options (use pointers & structures)

- (i) Listing names, dob sorted on names
- (ii) Listing names, dob sorted on dob
- (iii) Listing names, dob sorted on names, dob

11. a) Write a C program To copy the one file contents to the another file (using commandline arguments)

b) Write a C Program to count the frequencies of words in a given file.

Course Code: CS1703

Course Name: IT Workshop Lab

No. of Credits: 2

Detailed Syllabus:

1. Identifying the components or parts of MotherBoard.

2.a.Installation process of windows-7		
b.Installation process of ubuntu		
3.Regarding to Latex.		
a.Write Fresher Resume for Engineering students using Latex code.		
b.Prepare a document using Latex commands.		
c.Write a Latex commands for mathematical formulae.		
d.Construct a table using Latex commands.		
e.Construct your own set of five equations using Latex Commands.		
4.Regarding to Beamer.		
a.Prepare presentation of Resume using beamer.		
b.Write a Latex code adding graphics/image in Beamer.		
5.Regarding to Sci lab.		
a.Write a code for VectorOperations,MatrixOperations,Scripts and Functions,ConditionalBranching using Scilab		
b.Write a code for Iteration,Plotting2D graphics,10Xcos using Scilab.		
c.Write a code for File handling,Userdefined input and output,integration.		
d.Write a code for Solving Nonlinear equations,Linear equations,Linear equations using iterative Methods.		
e.Write a code for Interpolation,ODE Euler method,ODE Appllication,Optimization using karmarkar.		
f.Write a code for Control systems,Discrete systems,Calling user Defined Functions in Xcos.		
<u>E1 Sem2 Courses and Syllabus</u>		
Course Code:	Course Name: Basic Electronics	No. of Credits: 4
Detailed Syllabus:		
Course Code:	Course Name: Probability and Statistics	No. of Credits: 4
UNIT 1 :		
Algebra of Sets and Counting Methods , Introduction to uncertainty , Definitions of Probability , Theorems in Probability , Bayes‘Theorem and Its Applications		
UNIT 2 :		

Random Variable , Bivariate random variable , Mathematical Expectation , Discrete Probability Distributions , Continuous Probability Distributions , Functions of Random Variables , Correlation coefficient and Bivariate Normal Distribution

UNIT 3 :

Probability Inequalities and Generating Functions , Moment Generating Function , Characteristic Function , Cumulant Generating Function , Probability Generating Function

UNIT 4 :

Order Statistics , Convergence of Sequence of Random Variables , Weak Law of Large Numbers , Strong Law of Large Numbers , Central Limit Theorem.

UNIT5:

Linear and curvilinear regression: Linear regression, Regression coefficient its properties. Angle between

two lines of regression. Standard error of estimate or residual variance. Correlation coefficient between observed and estimated values. Curvilinear regression, Regression curves.

UNIT-6:

Large Sample theory: Introduction. Types of Sampling. Parameter and Statistic. Tests of Significance. Procedure for Testing of Hypothesis. Tests of significance for Large Samples. Sampling of Variables. Sampling of Attributes.

Course Code: CS1202	Course Name: Data structures	No. of Credits: 4
----------------------------	-------------------------------------	--------------------------

Detailed Syllabus: Algorithm Analysis: Mathematical Back Ground, Model, What to Analyze, Running Time

Calculations. Lists: Abstract Data Types, The List ADT, Singly Linked List ADT, Doubly Linked List ADT,

Circular Linked List ADT, Polynomial ADT. Stacks and Queues: The Stack ADT and its applications such as Infix to Postfix expression conversions, Evaluation of Postfix expressions, Delimiter Matching. The Queue ADT, The Circular Queue ADT. Sorting Preliminaries – Shellsort – Mergesort – Quicksort. Preliminaries – Binary Trees – Implementation, Expression trees. The Search Tree ADT – Binary Search Trees, Implementation. AVL Trees – Single Rotations, Double rotations. Hashing – General Idea – Hash Function – Separate Chaining – Open Addressing – Linear Probing – Priority Queues (Heaps) – Model – Simple implementations – Binary Heap – Heap Sort. Graphs: Definitions, representations, graph traversals.

Course Code: CS1203	Course Name: Introduction to Computer Science – 2 (C++)	No. of Credits: 4
----------------------------	--	--------------------------

Detailed Syllabus:

UNIT I: An Overview of C++: The Origins of C++, What is Object Oriented Programming, some C++ fundamentals, Old-Style Vs Modern C++, Introducing C++ Classes, Function

Overloading, Operator Overloading, Inheritance, Constructors and Destructors, The C++ Keywords, The General Form of a C++ Program

Classes and Objects: Classes, Structures and Classes, Unions and Classes are Related, Friend Functions, Friend Classes, Inline Functions, Parameterized Constructors, Static Class Members, When Constructors and Destructors are Executed, Scope Resolution Operator, Nested Classes, Local

Classes, Passing and Returning Objects, Object Assignment

Arrays, Pointers, References and the Dynamic Allocation: Arrays of Objects, Pointers, References, Dynamic Allocation Operators, The Placement Forms of new and delete.

Unit-II

Function Overloading, Copy Constructors and Default Arguments: Function Overloading, Overloading Constructor Functions, Copy Constructors, Finding the Address of an Overloaded Function, Overload Anachronism, Default Arguments, Function Overloading and Ambiguity.

Operator Overloading: Creating Member Operator Function, Overloading Using a Friend Function,

Overloading new delete, Overloading Special Operators & Comma Operator

Inheritance: Base-Class Access Control, Inheritance and protected members, Inheriting Multiple Base Classes, Constructors, Destructors and Inheritance, Granting Access, Virtual Base Classes.

Unit-III

Virtual Functions & Polymorphism: Virtual Functions, The Virtual Attribute is inherited, Virtual

Functions are Hierarchical, Pure Virtual Functions, Using Virtual Functions, Early Vs Late Binding.

Templates: Generic Functions, Applying Generic Functions, Generic Classes, Typename and export

Keywords, Power of Templates.

Exception Handling: Fundamentals, Derived-Class Exceptions, Options, Terminate() and unexpected(), uncaught_exception(), exception and bad_exception Classes, Applying Exception Handling.

Unit – IV

The C++ I/O System Basics: Old Vs. Modern C++ I/O, Streams, Stream Classes, Formatted I/O,

Overloading << and >>, Creating Manipulators.

C++ File I/O: File Classes, Opening and Closing a File, Text Files, Unformatted Binary I/O, get(),

Getline() functions, Detecting EOF ignore() peak() putback() flush(), Random Access, I/O Status,

Customized I/O and Files.

Runtime Type ID and the Casting Operators: RTTI, Casting Operators, Dynamic_cast, Reinterpret_cast.

Namespaces, Conversion Functions and other Advanced Topics: Namespaces, The std Namespace, Creating Conversion Functions, const Member Functions and mutable, Volatile Member Functions, Explicit Constructors, asm Keyword, Linkage Specification, Array-Based I/O, Dynamic Arrays, Binary I/O with Array-Based Streams, Differences between C and C++. Introducing Standard Template Library: An Overview of STL

Course Code:	Course Name: Basic Electronics Lab	No. of Credits: 2
---------------------	---	--------------------------

Detailed Syllabus:

Course Code: CS1802	Course Name: Data structures Lab	No. of Credits: 2
----------------------------	---	--------------------------

Detailed Syllabus:

1. Code the following list ADT operations using array, single linked list, double linked list.

- (a) void is_emptyList(List 1)
- (b) List makeNullList(size n)
- (c) Position firstPost(List 1)
- (d) Position endPost(List 1)
- (e) Position nextPost(List 1, Position p)
- (f) Position prevPos(List 1, position p)
- (g) Position find(List 1, Element x)
- (h) Position findKth(List 1, int k)
- (i) void insert(List 1, Position p)
- (j) void delete(List 1, Position p)
- (k) void append(List 1, Element x)
- (l) int cmp(List 1, Position p1, Position p2)
- (m) int cmp2(List11, List12, Position p1, Position p2)
- (n) void swap(List 1, Position p1, Position p2)
- (o) Element retrieveElement(List 1, Position p)
- (p) void print element(List 1, Position p)

2 Using the above List ADT operations, Write a menu driven program to support following higher level list operations:

- (a) Create null list
- (b) Read a list of elements into the list.

- (c) Insert an element in the Kth position of the list
 - (d) Delete an element in the Kth position of the list
 - (e) Delete a given element from the list
 - (f) Find whether given element is present in the list
 - (g) Display the elements of the list
3. Write a program that reads two lists of elements, prints them, reverses them, prints the reverse list, sort the lists, print the sorted lists, merges the list, prints merge list.
 4. Implement a polynomial ADT and write a program to read two polynomials and print them, adds the polynomials, prints the sum, multiply the polynomials and print the product.
 5. Implement stack ADT and write a program that reads an infix arithmetic expression of variables, constants, operators (+, -, *, /) and converts it into the corresponding postfix form. Extend the program to handle parenthesized expression also.
 6. Implement Queue ADT and write a program that performs Radix sort on a given set of elements.
 7. Implement the following sorting operations:-
 - (a) Shell Sort,
 - (b) Heap Sort
 - (c) Merge Sort
 - (d) Quick Sort
 8. Implement Binary Tree ADT and write a program that reads postfix Arithmetic expression form, builds the expression tree and performs tree Traversal on it.
 9. Implement Binary search ADT and write a program that interactively allows
 - (a) Insertion (b) Deletion (c) Find_min (d) Find_max (e) Find operations
 10. Implement AVL Tree ADT and Write a program that interactively allows
 - (a) Insertion (b) Deletion (c) Find_min (d) Find_max
 11. Implement Hashing and Write a program to find a element using Open Addressing.

Course Code: CS1803

Course Name: Introduction to Computer
Science - 2 Lab

No. of Credits: 2

Detailed Syllabus:

1. program using functions functions with default arguments
implementation of call by value, address, reference

2. simple classes for understanding objects, member functions & constructors classes with primitive data members, classes with arrays as data members classes with pointers as data members

classes with constant data members classes with static member functions

3. compile time polymorphism ,operator overloading ,function overloading

4. run time polymorphism

inheritance

virtual functions

virtual base classes

templates

5. file handling

sequential access

random access

E2 Sem1 Courses and Syllabus**Course Code: CS2102****Course Name: Object Oriented
Programming (JAVA)****No. of Credits: 4****Detailed Syllabus:**

Introduction: Introduction to java, data types, dynamic initialization, scope and life time, operators, control statements, arrays, type conversion and casting, finals & blank finals. Classes and Objects : Concepts, methods, constructors, usage of static, access control, this key word, garbage collection, overloading, parameter passing mechanisms, nested classes and inner classes. Inheritance: Basic concepts, access specifiers, usage of super key word, method overriding, final methods and classes, abstract classes, dynamic method dispatch, Object class. Interfaces: Differences between classes and interfaces, defining an interface, implementing interface, variables in interface and extending interfaces. Packages: Creating a Package, setting CLASSPATH, Access control protection, importing packages. Strings: Exploring the String class, String buffer class, Command-line arguments. Library: Date class, Collection, Enumerations and Wrapper classes.

Exception Handling: Concepts of Exception handling, types of exceptions, usage of try, catch, throw, throws and finally keywords, Built-in exceptions, creating own exception sub classes.

Multithreading : Concepts of Multithreading, differences between process and thread, thread life cycle, Thread class, Runnable interface, creating multiple threads, Synchronization, thread priorities, inter thread communication, daemon threads, deadlocks, thread groups. I/O Streams: Streams, Byte streams, Character streams, File class, File streams. Applets: Concepts of Applets, life cycle of an applet, creating applets, passing parameters to applets, accessing

remote applet, Color class and Graphics
 Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling events. AWT: AWT Components, windows, canvas, panel, File Dialog boxes, Layout Managers, Event handling model of AWT, Adapter classes, Menu, Menu bar. Swing-I – swings introduction, JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons. Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Course Code: CS2103	Course Name: Algorithms	No. of Credits: 4
----------------------------	--------------------------------	--------------------------

Detailed Syllabus:

Introduction: Algorithm Design paradigms – motivation, concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations. Divide and Conquer: Structure of divide and conquer algorithms: examples, quick sort, Strassen Multiplication; Analysis of divide and conquer run time recurrence relations.

Greedy Programming: Overview of the greedy paradigm examples of exact optimization solution, Approximate solution (Knapsack problem) Shortest-Path Algorithms – Unweighted Shortest Paths – Dijkstra's Algorithm – Minimum Spanning Tree – Prim's and Kruskal's algorithms. Dynamic Programming: Overview, difference between dynamic programming and divide and conquer, Applications: Shortest path in graph, Matrix multiplication, Traveling Salesman Problem, longest Common sequence. Graph Searching and Traversal: Overview, Traversal methods (depth first and breadth first search), Applications of DFS – connected components, Bi-connected components. Back tracking: Overview, 8-queen problem and Knapsack problem.

Branch and Bound: LC searching Bounding, FIFO branch and bound, LC branch and bound application: 0/1 Knapsack problem, Travelling Salesman Problem.

Computational Complexity: Complexity measures, Polynomial Vs Non-polynomial time complexity; NP-hard and NP-complete classes, examples.

Course Code: CS2201	Course Name: Switching Circuits and Logic Design	No. of Credits: 4
----------------------------	---	--------------------------

Detailed Syllabus:

1. Switching Circuits: Logic families: TTL, nMOS, CMOS, dynamic CMOS and pass transistor logic (PTL) circuits, inverters and other logic gates, area, power and delay characteristics, concepts of fan-in, fan-out and noise margin. 2. Switching theory: Boolean algebra, logic gates, and switching functions, truth tables and switching expressions, minimization of completely and incompletely specified switching functions, Karnaugh map and Quine-McCluskey method, multiple output minimization, representation and manipulation of functions using BDDs, two-

level and multi-level logic circuit synthesis. 3. Combinational logic circuits: Realization of Boolean functions using NAND/NOR gates, Decoders, multiplexers. Logic design using ROMs, PLAs and FPGAs. Case studies 4. Sequential circuits: Clocks, flip-flops, latches, counters and shift registers, finite-state machine model, synthesis of synchronous sequential circuits, minimization and state assignment, asynchronous sequential circuit synthesis. 5. ASM charts: Representation of sequential circuits using ASM charts, synthesis of output and next state functions, data path control path partition-based design.

Course Code: CS1301

Course Name: Data Base Management Systems

No. of Credits: 4

Detailed Syllabus:

Introduction: General introduction to database systems; Database-DBMS distinction, Approaches to building a database, Data models, Database management system, Three-schema architecture of a database, Challenges in building a DBMS, Various components of a DBMS. E/R Model: Conceptual data modeling - motivation, Entities, Entity types, various types of attributes, Relationships, Relationship types, E/R diagram notation, Examples. Relational Data Model Concept of relations: Schema-instance distinction, Keys, referential integrity and foreign keys,

- 1) Relational algebra operators: Selection, Projection, Cross product, Various types of joins, Division, Example queries,
- 2) Tuple relation calculus,
- 3) Domain relational calculus,
- 4) Converting the database specification in E/R notation to the relational schema. SQL: Introduction, Data definition in SQL, Table, key and foreign key definitions, Update behaviors. Querying in SQL, Basic select- from- where block and its semantics, Nested queries - correlated and uncorrelated, Notion of aggregation, Aggregation functions group by and having clauses, Embedded SQL. Dependencies and Normal forms: Importance of a good schema design, Problems encountered with bad schema designs, Motivation for normal forms, dependency theory -functional dependencies, Armstrong's axioms for FD's, Closure of a set of FD's, Minimal covers, Definitions of 1NF, 2NF, 3NF and BCNF, Decompositions and desirable properties of them, Algorithms for 3NF and BCNF normalization, Multi-valued dependencies and 4NF, Join dependencies and definition of 5NF. Data Storage and Indexes: File organizations, Primary, Secondary index structures, Various index structures - hash-based, Dynamic hashing techniques, Multi-level indexes, B+ trees. Transaction processing and Error recovery: Concepts of transaction processing, ACID properties, Concurrency control, Locking based protocols for CC, Error recovery and logging, Undo, Redo, Undo-redo logging and recovery methods

Course Code: CS2702

Course Name: Object Oriented

No. of Credits: 2

	Programming Lab	
Detailed Syllabus: 1. Write a java program to demonstrate static member, static method and static block. 2. Write a java program to demonstrate method overloading and method overriding. 3. Write a java program to demonstrate finals, blank finals, final methods, and final classes. 4. Write a java program to demonstrate synchronous keyword. 5. Write a java program to implement multiple inheritance. 6. Write a program to demonstrate packages. 7. Write a java program to crate user defined exception class and test this class. 8. Write am applet program to demonstrate Graphics class. 9. Write GUI application which uses awt components like label, button, text filed, text area, choice, checkbox, checkbox group. 10. Write a program to demonstrate MouseListener, KeyboardListener, ActionListener, ItemListener. MouseMotionListener, 11. Develop swing application which uses JTree, Jtable, JComboBox. 12. Write a JDBC Application to implement DDL and DML commands. 13. Write a program to implement client/server applications using connection oriented & connection less mechanisms.		
Course Code: CS2801	Course Name: Switching Circuits and Logic Design Lab	No. of Credits: 2
Detailed Syllabus: Design AND, OR and EX_ OR gates using Nand (7400) gates and verify them. A. Design a BCD to 6-3-1-1 Code converter and verify. B. Design a 6-3-1-1 to Gray Code converter and verify. 2. Design a full adder circuit using AND, OR and XOR gates. Verify it. 3. Design a 4 - Bit comparator using logic gates. 4. Design a Pseudo-random bit generator and check its performance. 5. Design a 4 - bit ripple carry adder, and verify by adding unsigned and signed integers. Check the overflow condition. Use BCD-to-SSD Decoders to demonstrate the results. 6. Design a Master - Slave J-K Flip-Flop using Logic gates. A. Design a Bi - directional counter using J-K Flip-flops. B. Design a Counter which counts the following arbitrary sequence: 0101, 0001, 1000, 1001, 1010, 0000, 0101... 7. Design a priority multiplexer for 8 Devices. Each device has one data output line, a request output line, and an acknowledgement input line. The data from the highest priority device has to be made available at the output of the priority multiplexer, and an acknowledgement has to be sent to that device. Hint: The circuit may be designed using		

priority encoder, multiplexer and decoder.

8. Write and verify a VHDL code, using verilog, for simulation of a 4- Bit fast look ahead carry adder using Logic gates, full adders etc.
9. Write and verify a VHDL code, using verilog, for simulation of an 8-bit signed integer multiplier using carry save adders.

Course Code: CS2703

**Course Name: Data Base Management
Systems Lab**

No. of Credits: 2

Detailed Syllabus:

1. Create tables with all constraints and table options.
2. Perform all DDL & DML commands on tables with different type of syntax use
3. Joins
4. Subquery
 - A) The Subquery as Scalar Operand and use different types of operators subquery
 - B) Subqueries with ALL, ANY, IN, or SOME, EXISTS or NOT EXISTS
 - C) Row Subqueries & FROM Clause Subqueries
 - D) Correlated Subqueries with SET operators.
 - E) Rewriting Subqueries as Joins
 - F) Delete & UPDATE with Subquery
5. Views
6. Stored Programs with Compound Statement(Procedure, Function & Trigger)
7. Functions
 - A) Control Flow Functions
 - B) String Functions (Note: Any TEN Function)
 - C) Numeric Function (Note: Any Five Function)
 - D) Date & Time Function (Note: Any TEN Function)

Note: Dateformat with different specifier

- E) Full Text Search Function

F) Cast Function G) Bit Function H) Encryption and Compression Functions (Note: Any Five Function) I) Aggregate Functions (Note: Any Ten Function using in GROUP BY CLAUSES)		
<u>E2 Sem2 Courses and Syllabus</u>		
Course Code: CS2202	Course Name: Formal Languages and Automata Theory	No. of Credits: 4
Detailed Syllabus: Automata: Introduction to Automata, The central concepts of automata theory - Alphabets, Strings, Languages. Finite Automata: An Informal picture of finite automata, Deterministic finite automata (DFA) - Definition of DFA, DFA processing strings, Notations for DFA, Extended transition function, the language of DFA, Non deterministic finite automata (NFA) – Definition of NFA, Extended transition function, the language of NFA, Equivalence of DFA and NFA Finite Automata with ϵ transitions: Use of ϵ - transition, notation for an ϵ - NFA, Epsilon closures, extended transitions and languages, Applications. Regular Expressions and Languages: Regular expressions, finite automata and regular expressions, Algebraic laws of regular expressions. Properties of Regular Languages: Proving languages are not regular – Pumping lemma for regular languages, Applications of the pumping lemma, Closure Properties of Regular Languages, Equivalence and minimization of automata – Minimization of DFA (Construction based treatment & proofs are excluded) Context Free Grammars: Context Free Grammars, Parse Trees, Constructing parse trees, derivations and parse trees, ambiguous grammars. Pushdown Automata: Definition of the Pushdown automata, the languages of PDA, Equivalences of PDA's and CFG's. Context free languages: Normal form's for context- Free grammars, the pumping lemma for context free languages. Properties of Context free languages: closure properties for context free languages, Decision properties for CFL's. Introduction to Turing Machines: The Turing Machine, programming techniques for Turing machines. Undecidability: a language that is not recursively enumerable, an undecidable problem that is RE, Undecidability problems about TM, Post's Correspondence problem.		

Course Code: CS2203	Course Name: Introduction to Software Engineering	No. of Credits: 4
Detailed Syllabus: 1. Introduction: Software development life cycle; Process models 2. Requirements and Specifications: software requirements specification, formal requirements specification and verification axiomatic and algebraic specifications 3. Software design: function-oriented software design, object-oriented design, UML, design patterns, user interface design 4. Structured programming and implementation: Testing; Verification and validation, Software Metrics, Software Project management, coding and unit testing, integration and systems testing, debugging techniques, software quality SEI CMM and ISO- 9001. software reliability and fault-tolerance, software project planning, monitoring, and control, software maintenance, computer-aided software engineering (CASE), software reuse, component-based software development, extreme programming.		
Course Code: CS2204	Course Name: Unix and Shell Programming	No. of Credits: 4
Detailed Syllabus: Introduction to Unix:- Architecture of Unix, Features of Unix , Unix Commands – PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip. Unix Utilities:- Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, unlink, du, df, mount, umount, find, unmask, ulimit, ps, w, finger, arp, ftp, telnet, rlogin. Text processing utilities and backup utilities , detailed commands to be covered are tail, head , sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm, cmp, diff, tr, awk, cpio Introduction to Shells : Unix Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, Command-Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment Customization. Filters : Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count characters, Words or Lines, Comparing Files. Grep : Operation, grep Family, Searching for File Content. Sed : Scripts, Operation, Addresses, commands, Applications, grep and sed. awk: Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays,		

String Functions, String Functions, Mathematical Functions, User – Defined Functions, Using System commands in awk, Applications, awk and grep, sed and awk.

Interactive Korn Shell :

Korn Shell Features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, Options, Startup Scripts, Command History, Command Execution Process.

Korn Shell Programming :

Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

Interactive C Shell :

C shell features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, On-Off Variables, Startup and Shutdown Scripts, Command History, Command Execution Scripts.

C Shell Programming :

Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

File Management :

File Structures, System Calls for File Management – create, open, close, read, write, lseek, link, symlink, unlink, stat, fstat, lstat, chmod, chown, Directory API – opendir, readdir, closedir, mkdir, rmdir, umask.

Course Code: CS3101

Course Name: Computer Organization
and Architecture

No. of Credits: 4

Detailed Syllabus:

1. **Basic functional blocks of a computer:** CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study instruction sets of some common CPUs. 2. **Data representation:** signed number representation, fixed and floating point representations, character representation. Computer arithmetic integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication shift-and-add, Booth multiplier, carry save multiplier, etc. Division non-restoring and restoring techniques, floating point arithmetic. 3. **CPU control unit design:** hardwired and micro-programmed design approaches, Case study design of a simple hypothetical CPU. 4. **Memory system design:** semiconductor memory technologies, memory organization. 5. **Peripheral devices and their characteristics:** Input-output subsystems, I/O transfers program controlled, interrupt driven and DMA, privileged and non privileged instructions, software interrupts and exceptions. Programs and processes role of interrupts in process state transitions. 6. **Performance enhancement**

techniques: Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.
7. Memory organization: Memory inter-leaving, concept of hierarchical memory organization, cache memory, cache size vs block size, mapping functions, replacement algorithms, write policy.

Course Code: CS2803	Course Name: Introduction to Software Engineering Lab	No. of Credits: 2
----------------------------	--	--------------------------

Detailed Syllabus:

1. **Introduction:** Software life cycle, Concept of process in software engineering, Project development and project management processes and their characteristics
2. **Project Development:** Requirement specifications, High level architecture, Design techniques - Function and object oriented design, Coding and verification, Refactoring, Testing and defect analysis
3. **Project Management:** Effort estimation, Scheduling. Configuration and risk management, Project Monitoring metrics

Course Code: CS2802	Course Name: Unix and Shell Programming lab	No. of Credits: 2
----------------------------	--	--------------------------

Detailed Syllabus:

1. Working with different Unix commands, Pipes, I/O redirection.
2. Shell scripting for number patterns
3. Programs related to File handling in unix
4. Programs related to finding, searching of files etc..
5. Programs related to string handling
6. Programs related to awk, grep, fgrep etc

Course Code: CS3701	Course Name: Computer Organization and Architecture Lab	No. of Credits: 2
----------------------------	--	--------------------------

Detailed Syllabus:

1. Assignments on assembly language programming
2. Experiments on synthesis / design of simple data paths and controllers
3. Assignments on interfacing devices and systems like data acquisition systems

E3 Sem1 Courses and Syllabus

Course Code: CS3102	Course Name: Principles of Programming Languages	No. of Credits: 4
----------------------------	---	--------------------------

Detailed Syllabus:

1. **Imperative Languages:** block structure, scope rules, parameter passing, constructs like coroutines, tasks etc.
2. **Functional programming:** functions, recursion, macros, user-defined

control constructs, higher order constructs, types, data abstraction, polymorphism, semantics, implementation issues. 3. **Declarative programming:** declarative programming, Horn clauses, procedural interpretation of Horn clauses, SLD-resolution including unification, the logical variable, implementation issues: abstract machines and compiling to abstract machines. 4. **Object-oriented programming:** objects and programming with objects, classes and instances, hierarchies and inheritance, encapsulation, semantics of OO languages and implementation issues.

Course Code: CS3202

Course Name: Compilers

No. of Credits: 4

Detailed Syllabus:

1. **Overview of different phases of a compiler:** front-end; back-end; Lexical analysis: specification of tokens, recognition of tokens, input buffering, automatic tools
2. **Syntax analysis:** context free grammars, top down and bottom up parsing techniques construction of efficient parsers, syntax-directed translation, automatic tools
3. **Semantic analysis:** declaration processing, type checking, symbol tables, error recovery; Intermediate code generation: run-time environments, translation of language constructs
4. **Code generation:** flow-graphs, register allocation, code-generation algorithms; Introduction to code optimization techniques.

Course Code: CS3105

Course Name: Artificial Intelligence

No. of Credits: 4

Detailed Syllabus:

Introduction to AI and intelligent agents. **Problem Solving:** Solving Problems by Searching, heuristic search techniques, constraint satisfaction problems, stochastic search methods. **Game Playing:** minimax, alpha-beta pruning. **Knowledge and Reasoning:** Building a Knowledge Base: Propositional logic, first order logic, situation calculus. Theorem Proving in First Order Logic. Planning, partial order planning. Uncertain Knowledge and Reasoning, Probabilities, Bayesian Networks. Learning : Overview of different forms of learning, Learning Decision Trees, Neural Networks Introduction to Natural Language Processing.

Course Code:

Course Name: Free Elective

No. of Credits: 4

Detailed Syllabus:

Course Code:

Course Name: Elective

No. of Credits: 4

Detailed Syllabus:**Course Code: CS3802****Course Name: Compilers Lab****No. of Credits: 2****Detailed Syllabus:**

1. Familiarity with compiled codes (assembly language) of RISC and CISC machines
2. Writing a scanner, writing predictive parser for a small language
3. Small experiment with scanner (lex/flex) and parser (yacc/byson) generator (such as translation of regular expression to NFA or the construction of parse tree)
4. Writing scanner-parse specification for a small language, translation of the language to an intermediate form (e.g. three-address code)
5. Generation of target code (in assembly language).
6. Code improvement (optional).

E3 Sem2 Courses and Syllabus**Course Code: CS4101****Course Name: Computer Networks****No. of Credits: 4****UNIT – I**

Introduction : OSI, TCP/IP and other networks models, Examples of Networks: Novell Networks, Arpanet, Internet, Network Topologies WAN, LAN, MAN.

UNIT - II

Physical Layer : Transmission media copper, twisted pair wireless, switching and encoding asynchronous

communications; Narrow band, broad band ISDN and ATM.

UNIT - III

Data link layer : Design issues, framing, error detection and correction, CRC, Elementary Protocol-stop

and wait, Sliding Window, Slip, Data link layer in HDLC, Internet, ATM.

UNIT - IV

Medium Access sub layer : ALOHA, MAC addresses, Carrier sense multiple access. IEEE 802.X

Standard Ethernet, wireless LANs. Bridges

UNIT - V

Network Layer : Virtual circuit and Datagram subnets-Routing algorithm shortest path routing,

Flooding,
 Hierarchical routing, Broad cast, Multi cast, distance vector routing.
 UNIT – VI
 Dynamic routing – Broadcast routing. Rotary for mobility. Congestion, Control Algorithms – General
 Principles – of Congestion prevension policies. Internet working: The Network layer in the internet and in the ATM Networks.
 UNIT –VII
 Transport Layer: Transport Services, Connection management, TCP and UDP protocols; ATM AAL Layer Protocol.
 UNIT – VIII
 Application Layer – Network Security, Domain name system, SNMP, Electronic Mail; the World WEB, Multi Media

Course Code: CS3202	Course Name: Operating Systems	No. of Credits: 4
----------------------------	---------------------------------------	--------------------------

Detailed Syllabus:

1. **Basics:** Operating System Functionalities, Types of Operating Systems, Computer Architecture support to Operating Systems 2. **Process Management:** Process Scheduling - Uniprocessor scheduling algorithms, Multiprocessor and Real-time scheduling algorithms, Process Synchronization - Peterson's Solution, Bakery Algorithm, Hardware Support to Process Synchronization, Semaphores, Critical Regions, Monitors - Deadlock prevention, deadlock avoidance and Deadlock Detection and Recovery - Bankers Algorithm, Threads 3. **Memory Management:** Segmentation and space allocation, Basics of linking and loading, Demand Paging, Page replacement algorithms, Analysis of page allocation policies - Working Set 4. **File Systems:** Contiguous, Sequential and Indexed Allocation, File system interface, File System implementation, Case study of Unix File system - Buffer Cache, Inodes, The system calls - ialloc, ifree, namei, alloc and free, Mounting and Unmounting files systems, Network File systems 5. **I/O System:** Disk Scheduling, Device drivers - block and character devices, streams, Character and Block device switch tables 6. **Protection and Security** - Accessibility and Capability Lists

Course Code:	Course Name: Free Elective	No. of Credits: 4
---------------------	-----------------------------------	--------------------------

Detailed Syllabus:

Applied Graph Theory

- 1 Fundamental concepts: basic definitions, operations, properties, proof styles
- 2 Trees: properties, distances and centroids, spanning trees, enumeration
- 3 Matchings: bipartite graphs, general graphs, weighted matching
- 4 Connectivity: vertex and edge connectivity, cuts, blocks, k-connected graphs, network flows
5. Traversability: Eulerian tours, Hamiltonian cycles); Coloring (vertex and edge coloring, chromatic number, chordal graphs
6. Planarity: duality, Euler's formula, characterization, 4-color theorem
7. Advanced topics: perfect graphs, matroids, Ramsay theory, extremal graphs, random graphs and Applications.

Text Books

1. Douglas B. West, Introduction to Graph Theory, Prentice Hall of India.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall.

Course Code:	Course Name: Elective	No. of Credits: 4
---------------------	------------------------------	--------------------------

Detailed Syllabus:

Web Techniques:

Designing web pages: HTML, forms, CGI scripts and clickable maps, JAVA applets, JAVAscript, JAVA servlets, Perl and PHP.

DHTML, XML.

Web security, Web services and Middleware

Course Code:	Course Name: Elective	No. of Credits: 4
---------------------	------------------------------	--------------------------

Detailed Syllabus:

Advanced ComputerArchitecture:

Overview of von Neumann architecture: Instruction set architecture; The Arithmetic and Logic Unit, The Control Unit, Memory and I/O devices and their interfacing to the CPU;

Measuringand reporting performance; CISC and RISC processors.

Pipelining: Basic concepts of pipelining, data hazards, control hazards, and structural hazards; Techniques for overcoming or reducing the effects of various hazards.

Hierarchical Memory Technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies.

Instruction-level parallelism: Concepts of instruction-level parallelism (ILP), Techniques for

increasing ILP; Superscalar, superpipelined and VLIW processor architectures; Vector and symbolic processors; Case studies of contemporary microprocessors Multiprocessor Architecture: Taxonomy of parallel architectures; Centralized shared-memory architecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, Cluster computers. Non von Neumann Architectures: Data flow Computers, Reduction computer architectures, Systolic Architectures.		
Course Code: CS4701	Course Name: Computer Networks lab	No. of Credits: 2
Detailed Syllabus: 1. Implement the data link layer framing methods such as character, character stuffing and bit stuffing. 2. Implement on a data set of characters the three CRC polynomials. 3. Implement Dijkstra 's algorithm to compute the Shortest path thru a graph. 4. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table art each node using distance vector routing algorithm 5. Take an example subnet of hosts . Obtain broadcast tree for it. 6. Take a 64 bit playing text and encrypt the same using DES algorithm . 7. Write a program to break the above DES coding 8. Using RSA algorithm Encrypt a text data and Decrypt the same		
Course Code: CS3801	Course Name: Operating Systems Lab	No. of Credits: 2
Detailed Syllabus: 1. Implementation of a command shell. 2. Implementation of Synchronization constructs. 3. Study of scheduling algorithms 4. Implementation of File system constructs 5. Advanced file system constructs		
E3 Summer Semester		
Course Code: CS3301	Course Name: Summer internship	No. of Credits: 6

<u>E4 Sem1 and Sem2 Courses are Electives and Projects</u>		
Course Code:	Course Name: Project-I	No. of Credits: 8
Course Code:	Course Name: Free Elective	No. of Credits: 4
Detailed Syllabus:		
Course Code:	Course Name: Elective-I	No. of Credits: 4
Detailed Syllabus:		
Object Oriented System Design:		
This course will cover object-oriented approach to modeling, problem solving, requirement analysis, system design, system implementation, database design, system engineering and software engineering.		
Fundamental concepts of Object Oriented Programming : Introduction to the principles of object-oriented programming (classes, objects, messages, encapsulation, inheritance, polymorphism, exception handling, and object-oriented containers).		
Object design implementation in a programming language, e.g, C++ or Java.		
Object oriented analysis, modeling and design. UML may be introduced.		
Use cases, Use case driven analysis.		
Structural Modeling : classes, relationships , interfaces, class diagrams, and object diagrams, in UML.		
Behavioural / Functional modelling: use case diagrams, sequence diagrams, in UML.		
Dynamic Modelling : state charts		
Architectural Modelling		
Analysis patterns, Design patterns.		
Distributed Object Model: CORBA and COM / DCOM		
Object oriented Database Systems: Object oriented Data model, query languages,		
Storage Organization and Indexing Techniques		
Object Relational Databases		

Course Code:	Course Name: Elective-II	No. of Credits: 4
Detailed Syllabus: Digital System Testing and Testable Design: Circuit Modeling - Basic Concepts, Functional Modeling at Logic and Register levels, Structural Models. Logic Simulation - Simulation based Design Verification, Delay Models, Gate level Event Drive Simulation. Fault Modeling - Logical Fault Models, Fault Detection, Equivalence and Dominance, Single and Multiple Stuck - Fault Model. Fault Simulation - General Fault Simulation Techniques, Fault Simulation for Combinational Circuits, Fault Sampling. Testing: Algorithms for Testing Single Stuck Fault and Bridge Faults, Automatic Test Generation Concepts, Functional Testing, Random Test Generators, Encoding techniques. Design for Testability: Scan Based Design, Boundary Scan Techniques, Compression Techniques, LFSRs, Built-in Self Test (BIST), BIST Architectures and Advanced BIST Concepts. Formal Verification: Model Checking, Equivalence Checking and Theorem Proving, Design of tools for Formal Verification.		
Course Code:	Course Name: Elective-III	No. of Credits: 4
Detailed Syllabus: Mobile Application Development: Principles and Design Introduction to Mobile Applications: Characteristics, Benefits, Overview of Available Technologies Mobile Application Design: Application Model and Infrastructure, Hardware and Software Architecture, Managing Resources, Development Workflow Development of Mobile Applications with MIDP and Symbian Mobile Graphics: User Interface, Scalable Vector Graphics, Mobile 3D Graphics (M3G) API Security: Secure Design, MIDP security features, Security for Symbian OS Networking: Basic Connectivity, Bluetooth Connectivity, Web Services Technology Integration		

Design, develop and deploy a commercial application for Mobile Phones.		
Course Code:	Course Name: Project-II	No. of Credits: 12
Course Code:	Course Name: Free Elective	No. of Credits: 4
Detailed Syllabus: Parallel Algorithms Parallel Models (SIMD, MIMD, PRAMs, Interconnection Networks), Performance Measures (Time, Processors, Space, Work). Interconnection Architectures (Linear Array, Meshes, Trees, Mesh of Trees, Hyper cubes, Butterfly Networks, Cube Connected Cycles, Benes Networks). Techniques (Balanced Trees, Pointer Jumping, Divide and Conquer, Partitioning, Pipelining, Systolic Computation, Accelerated Cascading, Prefix Computation, List Ranking, Euler Tour, Tree Contraction). Sorting, Searching, Merging; Matrix Operations. Graph Algorithms (Connected Components, Spanning Trees, Shortest Paths). Complexity (Lower bounds, NC Class and P-Completeness).		
Course Code:	Course Name: Elective-I	No. of Credits: 4
Detailed Syllabus: Network Security Conventional Encryption: Classical Techniques; Modern Techniques; Algorithms; Confidentiality Using Conventional Encryption. Public-Key Encryption and Hash Functions: Public-Key Cryptography; Introduction to Number Theory; Message Authentication and Hash functions; Hash and Mac Algorithms; Digital Signatures and Authentication Protocols. Network Security Practice: Authentication applications; Kerberos, X.509 Directory Authentication Service, Electronic Mail Security; Pretty Good Privacy, S/MIME, IP Security; IP Security Architecture, Combining Security Associations, Key Management,		

Web Security; Web Security Requirements, Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction. System Security: Intruders, Viruses and Related Threats. Firewalls; Firewall Design Principles, Trusted Systems.		
Course Code:	Course Name: Elective-II	No. of Credits: 4
Detailed Syllabus: Data mining: Types of data mining problems. The process of data mining. Statistical evaluation of big data: statistical prediction, performance measures, pitfalls in data-mining evaluation. Data preparation: data models, data transformations, handling of missing data, time-dependent data, textual data. Data reduction: feature selection, principal components, smoothing data, case subsampling. Predictive modeling: mathematical models, linear models, neural nets, advanced statistical models, distance solutions, logic solutions, decision trees, decision rules, model combination. Solution analyses: graphical trend analyses, comparison of methods. Case studies. Future trends: text mining, visualization, distributed data. Practical sessions using open-source software.		
Course Code:	Course Name: Viva	No. of Credits: 2
List of Electives will be offered during E4Sem1 and Sem2		
Course Code:	Elective1: Neural Networks and Applications	No. of Credits: 4
Detailed Syllabus: Neurons and neural networks, basic models of artificial neural networks: simple layer perception, feed forward multilayer perceptron, Hopfield networks, competitive learning networks, applications of neural networks for matrix algebra problems, adaptive filtering and adaptive pattern recognition, dynamic system identification, dynamic system modeling using recurrent neural networks, approximation/optimization problems, VLSI implementation of neural networks.		

Course Code:	Elective2: Multimedia Technology	No. of Credits: 4
Detailed Syllabus: Introduction to Multimedia System: Architecture and components, Multimedia distributed processing model, Synchronization, Orchestration and Quality of Service (QOS) architecture. Audio and Speech: Data acquisition, Sampling and Quantization, Human Speech production mechanism, Digital model of speech production, Analysis and synthesis, Psycho-acoustics, low bit rate speech compression, MPEG audio compression. Images and Video: Image acquisition and representation, Composite video signal NTSC, PAL and SECAM video standards, Bilevel Image Compression standards: ITU (formerly CCITT) Group III and IV standards, JPEC image compression standards, MPEG video compression standards. Multimedia Communication: Fundamentals of data communication and networking, Bandwidth requirements of different media, Real time constraints: Audio latency, Video data rate, multimedia over LAN and WAN, Multimedia conferencing. Hypermedia presentation: Authoring and Publishing, Linear and mon-liner presentation, Structuring Information, Different approaches of authoring hypermedia documents, Hypermedia data models and standards. Multimedia Information Systems: Operating system support for continuous media applications: limitations is usual OS, New OS support, Media stream protocol, file system support for continuous media, data models for multimedia and hypermedia information, content based retrieval of unstructured data		
Course Code:	Elective3: Pattern Recognition and Application	No. of Credits: 4
Detailed Syllabus: Pattern Recognition : Feature Extraction and classification stages, Different approaches to pattern recognition. Statistical Pattern Recognition . Hypothesis testing, Linear classifiers, Parametric and nonparametric classification techniques, Unsupervised learning and clustering, Syntactic pattern recognition, Fuzzy set Theoretic approach to PR, Applications of PR : Speech and speaker recognition, Character recognition, Scene analysis		