ANDHRA UNIVERSITY:: VISAKHAPATNAM
COMMON SCHEME OF INSTRUCTION & EXAMINATION
**I/IV B.TECH** (FOUR YEAR COURSE) &
**I/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)
(With effect from **2015-2016** admitted batch onwards)
Under Choice Based Credit System

**GROUP – A**

**(Civil, Chemical, CSE, IT)**

**I-SEMESTER**

| Code No. | Course | Credits | Lecture Hrs | Tutorial Hrs | Lab Hrs | Total Contact Hrs/Week | Sessional Marks | Exam Marks | Total Marks |
|---|---|---|---|---|---|---|---|---|---|
| ENG 1101 | English | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1102 | Mathematics-I | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1103 | Mathematics-II | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1104 | Chemistry | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1105 | Computer Programming with C and Numerical Methods | 4 | 2 | -- | 3 | 5 | 30 | 70 | 100 |
| ENG 1106 | History of Science and Technology | 2 | 2 | -- | -- | 2 | 30 | 70 | 100 |
| ENG 1107 | Chemistry Lab | 2 | -- | -- | 3 | 3 | 50 | 50 | 100 |
| ENG 1108 | Computer Programming with C and Num. Methods Lab | 2 | -- | -- | 3 | 3 | 50 | 50 | 100 |
| ENG 1109 | Sports(Audit) | -- | -- | -- | -- | 3 | -- | -- | -- |
| | **TOTAL** | **26** | **16** | **4** | **9** | **32** | **280** | **520** | **800** |

| ENG 1105 | Computer Programming using C& Numerical Methods | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

1. **Introduction to C:** Basic structure of C program, Constants, Variables and data types, Operators and Expressions, Arithmetic Precedence and associativity, Type Conversions. Managing Input and Output Operations Formatted Input, Formatted Output.
2. **Decision Making, Branching, Looping, Arrays & Strings:** Decision making with if statement, Simple if statement, The if…else statement, Nesting of if…else statement, the else..if ladder, switch statement, the (?:) operator, the GOTO statement., The while statement, the do statement, The for statement, Jumps in Loops ,One, Two-dimensional Arrays, Character Arrays. Declaration and initialization of Strings, reading and writing of strings, String handling functions, Table of strings.
3. **Functions:** Definition of Functions, Return Values and their Types, Function Calls, Function Declaration, Category of Functions: No Arguments and no Return Values, Arguments but no Return Values, Arguments with Return Values, No Argument but Returns a Value, Functions that Return Multiple Values. Nesting of functions, recursion, passing arrays to functions, passing strings to functions, the scope, visibility and lifetime of variables. .
4. **Pointers:** Accessing the address of a variable, declaring pointer variables, initializing of pointer variables, accessing variables using pointers, chain of pointers, pointer expressions, pointers and arrays, pointers and character strings, array of pointes, pointers as function arguments, functions returning pointers, pointers to functions, pointers to structures-Program Applications
5. **Structure and Unions:** Defining a structure, declaring structure variables, accessing structure members, structure initialization, copying and comparing structure variables, arrays of structures, arrays within structures, structures within structures, structures and functions and unions, size of structures and bit-fields- Program applications.
6. **File handling:** Defining and opening a file, closing a file, Input/ Output operations on files, Error handling during I/O operations, random access to files and Command Line Arguments- Program Applications
7. **Numerical Methods:** Solutions of Algebraic and Transcendental Equations**,** Bisection Method, Newton Raphson Method. Newton's forward and backward Interpolation, Lagrange's Interpolation in unequal intervals. Solutions of Linear Equations: Gauss Elimination Method, Gauss Jacobi and Gauss Seidel Methods. Numerical Integration: Trapezoidal rule, Simpson's 1/3 rules. Solutions of Ordinary First Order Differential Equations: Euler's Method, Modified Euler's Method and Runge-Kutta Method.

**Text Books:**


1. Programming in ANSI C, E Balagurusamy, 6th Edition. McGraw Hill Education (India) Private Limited.
2. Introduction to Numerical Methods, SS Sastry, Prentice Hall


**Reference Books:**


1. Let Us C , Yashwant Kanetkar, BPB Publications, 5th Edition.
2. Computer Science, A structured programming approach using C", B.A.Forouzan and R.F.Gilberg, " 3rd Edition, Thomson, 2007.
3. The C –Programming Language' B.W. Kernighan, Dennis M. Ritchie, PHI.
4. Scientific Programming: C-Language, Algorithms and Models in Science, Luciano M. Barone (Author), Enzo Marinari (Author), Giovanni Organtini, World Scientific.

ANDHRA UNIVERSITY: : VISAKHAPATNAM
COMMON SCHEME OF INSTRUCTION & EXAMINATION
**I/IV B.TECH** (FOUR YEAR COURSE) &
**I/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)
(With effect from **2015-2016** admitted batch onwards)
Under Choice Based Credit System

## GROUP – A
## (Civil, Chemical, CSE, IT)

## II-SEMESTER

| Code No. | Course | Credits | Lecture Hrs | Tutorial Hrs | Lab Hrs | Total Contact Hrs/Week | Sessional Marks | Exam Marks | Total Marks |
|---|---|---|---|---|---|---|---|---|---|
| ENG 1201 | Mathematics-III | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1202 | Physics | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1203 | Probability, Statistics and Queuing Theory | 4 | 3 | 1 | -- | 4 | 30 | 70 | 100 |
| ENG 1204 | Engineering Graphics | 4 | 4 | -- | -- | 4 | 30 | 70 | 100 |
| ENG 1205 | Ethics & Moral Values | 2 | 2 | -- | -- | 2 | 30 | 70 | 100 |
| ENG 1206 | Physics Lab | 2 | -- | -- | 3 | 3 | 50 | 50 | 100 |
| ENG 1207 | Work shop | 2 | -- | -- | 3 | 3 | 50 | 50 | 100 |
| ENG 1208 | English Language Lab | 2 | -- | -- | 3 | 3 | 50 | 50 | 100 |
| ENG 1209 | NCC/NSS (Audit) | -- | -- | -- | -- | 3 | -- | -- | -- |
| | **Total** | **26** | **15** | **3** | **9** | **30** | **300** | **500** | **800** |

| ENG 1203 | PROBABILITY, STATISTICS & QUEUING THEORY | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To discuss basics of probability and related theorems , Problems. To study about conditional probability and Bayes theorem.
2. To study about random variables and their properties. To examine , analyze and compare Probability distributions.
3. To discuss regression and estimation techniques.
4. To discuss various types of tests such as F-test, Chi-square test. To study the various queuing models.

**Course Outcomes:**

At the end of the course student will be able to

1. ability to solve various problems regarding probability and conditional probability.
2. Examine , analyze and compare probability distributions.
3. Prepare null and alternative hypothesis and test its validity based on random sample.
4. ability to solve various types of regression problems.
5. Ability to understand various queuing models.

## Syllabus:

1. **Probability:** Definitions of Probability, Addition Theorem, Conditional Probability, Multiplication Theorem, Bayes' Theorem of Probability and Geometric Probability.

2. **Random Variables and their Properties:** Discrete Random Variable, Continuous Random Variable, Probability Distribution, Joint Probability Distributions Their Properties, Transformation Variables, Mathematical Expectations, Probability Generating Functions.

3. **Probability Distributions:** Discrete Distributions: Binomial, Poisson Negative Binominal Distributions and their Properties; Continuous Distributions : Uniform, Normal, Exponential Distributions and their Properties.

4. **Multivariate Analysis and Curve Fitting**: Correlation, Correlation Coefficient, Rank Correlation, Regression Analysis, Multiple Regression, Principles of Least Squares and Curve Fitting

5. **Estimation and testing of hypothesis**: Sample, Populations, Statistic, Parameter, Sampling Distribution, Standard Error, Un-Biasedness, Efficiency, Maximum Likelihood Estimator, Notion & Interval Estimation.

6. **Sample Tests:** Large Sample Tests Based on Normal Distribution , Small Sample Tests : Testing Equality of Means, Testing Equality of Variances, Test of Correlation Coefficient, Test for Regression Coefficient; Coefficient of Association, $^2$ – Test for Goodness of Fit, Test for Independence.

7. **Queuing Theory** : Queue Description, Characteristics of a Queuing Model, Study State Solutions of M/M/1: Model, M/M/1 ; N Model, M/M/C: Model, Case Studies

**Text Books** :

1. Probability & Statistics for Engineers and Scientists,Walpole, Myers, Myers, Ye. Pearson Education.
2. Probability, Statistics and Random Processes T.Veerarajan Tata McGraw – Hill

**Reference Books:**

1. Probability & Statistics with Reliability, Queuing and Computer Applications, Kishor S. Trivedi, Prentice Hall of India ,1999

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A), VISAKHAPATNAM**

**I – SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION**

**Branch: COMPUTER SCIENCE AND ENGINEERING**

**II/IV B.TECH** (FOUR YEAR COURSE) &
**II/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)
(With effect from **2015-2016** admitted batch onwards)
Under Choice based Credit System

| **B.TECH. (CSE) 2ND YEAR I-SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION WITH EFFECT FROM 2015-2016** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **SUB.REF.** | **NAME OFTHE SUBJECT** | **PERIODS** | | | **MAXIMUM MARKS** | | | **CREDITS** |
| | | **THEORY** | **TUTORIAL** | **LAB** | **EXAM** | **SESSIONALS** | **TOTAL** | |
| CSE 2.1.1 | DATA STRUCTURES | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.1.2 | ELEMENTS OF ELECTRONICS ENGINEERING | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.1.3 | DISCRETE MATHEMATICAL STRUCTURES | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.1.4 | OBJECT ORIENTED PROGRAMMING | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.1.5 | ELEMENTS OF ELECTRICAL ENGINEERING | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.1.6 | DIGITAL LOGIC DESIGN | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.1.7 | DATA STRUCTURES LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 2.1.8 | OBJECT ORIENTED PROGRAMMING LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| **TOTAL CREDITS** | | | | | | | | **28** |

| CSE 2.1.1 | DATA STRUCTURES | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. Assess how the choice of data structures and algorithm design methods impacts the performance of programs.
2. Choose the appropriate data structure and algorithm design method for a specified application.
3. Solve problems using data structures such as linear lists, stacks, queues, binary trees, heaps binary search trees, and graphs and writing programs for these solutions.

**Course Outcomes:**

1. Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithm.
2. Demonstrate different methods for traversing trees.
3. Compare alternative implementations of data structures with respect to performance.
4. Discuss the computational efficiency of the principal algorithms for sorting and searching

**Syllabus:**

1. **Introduction to Data Structures**: Review of C Programming, Recursive Definition and Processes, Recursion in C, Simulation of Recursion, Efficiency of Recursion, Abstract Data Types, Meaning and Definition of Data Structures, Arrays

2. **Stacks**: Stack as an Abstract Data Type, Primitive Operations, Implementing Stack Operations using Arrays, Infix, Postfix and Prefix: Definitions, Evaluation and Conversions. **Queues**: Queue as an Abstract Data Type, Sequential Representation, Types of Queues, Operations, Implementation using Arrays.

3. **Linked List**: Operations, Implementation of Stacks, Queues and priority Queues using Linked Lists+, Circular Lists: Insertion, Deletion and Concatenation Operations, Stacks and Queues as Circular Lists, Doubly Linked Lists.

4. **Trees**: Binary Trees - Definitions and Operations, Binary Tree Representation: Node Representation, Implicit array Representation, Binary Tree Traversal, Threaded Binary Trees and their Traversal, Trees and their Applications; Tree Searching: Insertion and Deletion of a node from a Binary Search Tree, Efficiency of Binary Search Tree operations.

5. **Searching**: Basic Searching Techniques: Dictionary as an Abstract Data Type, Algorithmic Notation, Sequential Searching and its Efficiency, Binary Search, Interpolation Search.

6. **Sorting**: General Background: Efficiency, Asymptotic Notations, Efficiency of Sorting, Bubble Sort and Quick Sort and their Efficiency, Selection Sorting, Binary Tree Sort, Heap Sort, Insertion Sorts , Shell Sort , Address calculation Sort , Merge and Radix Sorts.

7. **Graphs and Their Application**: Definition of Graphs, Representation of Graphs, Transitive closure, Linked Representation of Graphs, Topological Ordering of nodes, Graph Traversal and Spanning Forests, Undirected Graphs and their Traversals, Applications of Graphs, Minimal Spanning Trees.

**Textbooks**:

1. Data Structures Using C and C++ Yddish Langsam, Moshe J. Augenstein and Aaron M.Tanenbaum, Prentice Hall Of India ($2^{nd}$ Edition)
2. Data Structures, Algorithms and Applications with C++, Sahani Mc-Graw Hill.

| CSE 2.1.2 | ELEMENTS OF ELECTRONICS ENGINEERING | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To introduce Electronics and Communication Engineering in a nutshell.
2. To explain the role of Electronics and Communication Engineering in all other engineering disciplines.
3. To explain the basic building blocks of digital and analog electronic circuits.

**Course Outcomes:**

At the end of the course, the student must be able to

1. Design simple combinational and sequential circuits.
2. Analyze the given RC and RL circuits.
3. Design simple Diode circuits like rectifiers and clipping circuits.
4. Design circuits using ideal opamp to perform mathematical operations on analog signals.
5. Appreciate the importance of some of the analog systems such as ADC, DAC.

**Syllabus:**

1. **Introduction to Electronics and Semiconductors:** Energy band theory, Conduction in Insulators, Semiconductors and metals, Electron emission from metals, Classification of semiconductors, Carrier concentration in an intrinsic semiconductor, Properties of intrinsic semiconductor, Drift and diffusion currents.
2. **Semi Conductor Diode :** Theory of PN junction diode, Open circuited PN junction, V-I characteristics of a PN diode, Diode current equation, Transition and diffusion capacitances , Break down in PN diode, Applications of PN diodes. Zener diode, Zener regulator, Tunnel diode , Schottky diode.
3. **Rectifying circuits:** Half wave and full wave rectifiers, Bridge rectifiers, Efficiency, Ripple and regulation of each rectifier , Capacitor filters.
4. **Bipolar Junction Transistor** :– Introduction, construction, Operation of PNP and NPN Transistors – Transistor Circuit configurations- Characteristics of a CE configurations – h parameters, low frequency small signal equivalent circuit of a Transistor.
5. **Transistor Biasing and thermal stabilization:** Transistor Biasing, Stabilization, Different methods of transistor biasing – Fixed bias, Collector feedback bias – self bias – Bias compensation.
6. **Transistor Amplifiers:** CE, CB, CC amplifier configurations –Multistage amplifier – A Two Stage RC coupled amplifier – frequency response curve and bandwidth**.**
7. **Field Effect Transistors:** Junction Field Effect Transistors (JFET) – JFET characteristics, JFET Parameters, Small signal equivalent circuit – MOSFETS – Depletion and Enhancement MOSFETS.

**Text Books:**

1. Electronic Device and Circuits by Sanjeev Guptha.

**Reference Books:**

1. Electronic Device and Circuits Theory by Robert L. Boylested Electronic Device and Circuits by David. A. Bell

| CSE 2.1.3 | DISCRETE MATHEMATICAL STRUCTURES | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To understand mathematical arguments using logical connectives and quantifiers and verify the validity of logical flow of arguments using propositional ,predicate logic and truth tables.
2. To understand about permutations and combinations.
3. To understand various types of relations and discuss various properties of the relations.
4. To study the graphs, graph isomorphism and spanning trees.
5. To study about Boolean algebra and Finite State Machines.

**Course Outcomes:**

At the end of the course student will be able to

1. Rewrite mathematical arguments using logical connectives and quantifiers and verify the validity of logical flow of arguments using propositional, predicate logic.
2. Identify and give examples of various types of relations and describe various properties of the relations.
3. Ability to solve problems using permutations and combinations.
4. Determine isomorphism of graphs and spanning tree of a given graph using BFS/DFS algorithms. Also determine minimal spanning tree of a given graph.

**Syllabus:**

1. **The Foundations-Logic and Proofs:** Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers Rules of Inference, Introduction to Proofs, Proof Methods and Strategy, Basic Structures-Sets, Functions, Sequences and Sums: Sets, Set Operations, Functions, Sequences and Summations.

2. **The Fundamentals-Algorithms, the Integers and Matrices:** Algorithms, The Growth of Functions, Complexity of Algorithms, The Integers and Division, Primes and Greatest Common Devisors, Integers and Algorithms, Applications of Number Theory, Matrices.

3. **Induction and Recursion:** Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness.
   **Counting:** The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations.

4. **Advanced Counting Techniques:** Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recursion Relations, Generating Functions, Inclusion-Exclusion, and Applications of Inclusion-Exclusion.

5. **Relations:** Relations and their properties, n-ary relations, applications, Representation, closure, equivalence relations, Partial orderings.
   **Graphs**: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring

6. **Trees:** Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees,

7. **Boolean Algebra:** Boolean Functions, Representing Boolean Functions, Logic Gates, Minimization of Circuits

   **Modeling Computation:** Languages and Grammars, Finite-State Machines with Output, Finite-State Machines with No Output, Language Recognition, Turing Machines

**Text Book:**

1. Discrete Mathematics & Its Applications with Combinatorics and Graph Theory by Kenneth H Rosen, Tata McGraw-Hill Publishing Company Ltd., New Delhi.

**Reference Books:**

1. Discrete Mathematics for Computer Scientists & Mathematicians by Joe L. Mott, Abraham Kandel, Theodore P. Baker, Prentice-Hall, India.

2. Discrete Mathematics by Richard Johnson Baug, Pearson Education, New Delhi.

3. Discrete and Combinatorial Mathematics by Ralph. G. Grimaldi, Pearson Education, New Delhi.

| CSE 2.1.4 | OBJECT ORIENTED PROGRAMMING | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

On completing this course student will be able to

1. Understand the syntax and principles of Object oriented programming language, and to programs using control statements, classes and interfaces.
2. Design and development of secure and extendable C++ applications.
3. Understanding the concepts of oops, different predefined classes and packages
4. Understand the concepts of polymorphism

**Course Outcomes :**

1. Students will be able to handle I/O streams and Run time errors.
2. Students will be able to construct applications and Identify where data structures are appearing in them

**Syllabus:**

1. **Basic Concepts of OOP:** Procedural Paradigms, Object Oriented Paradigm, OOP Principles and Terminology, OOP benefits, Procedure and Object Oriented programming languages, advantages and disadvantages. Introduction to U.M.L: Description of various U.M.L. Diagrams with examples.

2. **Introduction to C++ :** Basic Structure C++ Program, variable and Constants, Symbolic Constants , basic data types and derived data type, variable declaration, dynamic initialization, type modifiers, type casting, i/o statements in C++, operators and example programs**,** Control Structures- Programs using all control structures and statements, Functions: Function Prototypes, Function Components, Returning values from functions, actual and formal arguments, parameter passing methods, Inline functions,

3. **Classes and Objects:** Introduction to class, class definition, class specification, Member functions, data members, access specifiers, scope resolution operator, Object definition and creation, array of objects, pointers, Pointers to objects, this pointer, dynamic allocation operator, friend functions, const and volatile functions, static members, nested classes, local classes,

4. **Constructors and destructors:** Definition of constructor and destructor, default constructor, parameterized constructor, copy constructor, constructor with dynamic allocation, explicit constructor.
   **Inheritance:** Definition, base class, derived class, using access specifiers in inheritance, Types of Inheritance, protected data with private inheritance, constructor in derived and base class, abstract classes,

5. **Virtual functions and Polymorphism:** Function overloading, arrays and strings, Operator overloading through unary and binary operator, Friend functions, Assignment operator, Stream operator overloading and type conversion; Virtual functions, Pure Virtual function, Dynamic polymorphism, Virtual destructor, Virtual base class, Dynamic casting, Cross casting, Down casting, Program development.

6. **Streams and Files in C++:** Stream Classes, Formatted and unformatted data, manipulators, user defined manipulators, file streams, file pointer manipulation; file open and close, file handling, random access, object serialization, name spaces, std namespaces, ANSI string objects and standard template library.

7. **Templates, Exception handling:** Class templates, Function templates, Member function templates, Exception handling - try-catch-throw paradigm, exception specification, terminate and un expected functions- uncaught exception, exception handling mechanism, multiple catch, nested try, Rethrowing the exceptions

**Text Books:**

1. Object Oriented Programming through C++ by Robat Laphore.

**Reference Books:**

1. Object Oriented Programming in C++: N. Barkakati, PHI
2. Object oriented Programming using C++: E. Balagurusamy, PHI.
3. The Complete reference in C++ by Herbert Shieldt, TMH
4. The C++ Programming Language by B. Stroustrup, Pearson Education

| CSE 2.1.5 | ELEMENTS OF ELECTRICAL ENGINEERING | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

**1.** To provide exposure to basic electrical engineering concepts to non-major students.

**Course Outcomes:**

1. An ability to define and explain the meaning/function of charge, current, voltage, power, energy, resistors (R), and the fundamental principles of Ohm's law, KVL and KCL including an understanding of electrical safety.
2. An understanding of the behavior of inductances (L) and capacitances (C).
3. An ability to write the differential equations for a given RLC network and solve them analytically for the transient and steady state responses to a step input.
4. An ability to analyze resistive op amp circuits and design inverting, non-inverting, summing, and differential amplifier circuits using op amps.
5. An ability to qualitatively and quantitatively predict and compute the steady state AC responses of basic circuits using the phasor method

**Syllabus:**

1. **Introduction to Electrical Energy**: Definitions of magnetic circuit, Reluctance, Magneto-motive force), magnetic flux, Simple problems on magnetic circuits, Hysterisis, Characters and loss calculations, Faraday's laws of Electromagnetic Induction, Induced E.M.F., Dynamically induced E.M.F., Statistically Induced EMF, Self Inductance, Mutual Inductance.

2. **D.C. Generators**: D.C. Generator principle, Constructional details, E.M.F equation, Types and classification, Characteristics, Efficiency, Applications.

3. **D.C. Motors: D.C.** Principle and Operation, Significance of back E.M.F., Torque equation, Types, Speed control methods of D.C. Motors, Applications of D.C. Motor. Testing of D.C. Machines: Losses and Efficiency, Direct load test and Swinburne's test.

4. **A.C. Circuits**: Introduction to Steady State Analysis of A.C. Circuits, Series and Parallel R. l. and R.C. Circuits, Balanced 3 Phase Circuits, Star and delta connection

5. **Transformers:** Transformer principle, EMF equation of transformer, Transformer on load, Equivalent circuit of Transformer, Voltage regulation of Transformer, Losses in a Transformer, Calculation of Efficiency and Regulation by Open circuit and Short circuit Tests.

6. **Three phase Inductance Motor**: Construction of 3 Phase induction Motor, Principle of operation. Types of 3 phase induction Motor, Torque Equation of Induction Motor., slip – Torque characteristics., Starting Torque, Torque under running condition., Maximum Torque Equation., Power stages of Induction Motor., Efficiency Calculation of Induction Motor by direct loading.

7. **Alternator**: Principle of Operation, EMF equation of Alternator, Calculation of Voltage

Regulation by Sync, Impedance method, Synchronous Motor: Principle of Operation, Construction., Methods of starting of Synchronous Motor.

8. **Earthing:** Causes of High currents, Current diversion, Earthing principle, Types of Earthing, Earthing Process.

**Text Book**:

"Elements of Electrical Engineering and Electronics" by V.K.Mehta, S. Chand & Co

**Reference Book**:

"A First Course in Electrical Engineering" by Kothari.

| CSE 2.1.6 | DIGITAL LOGIC DESIGN | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

To introduce the basic principles for design of combinational circuit and sequential circuits. To learn simple digital circuits in preparation for computer engineering.

**Course Outcomes:**

A student who successfully fulfills the course requirements will have demonstrated:

1. An ability to define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
2. An ability to understand the different Boolean algebra theorems and apply them for logic functions.
3. 3. An ability to define the Karnaugh map for a few variables and perform an algorithmic reduction of logic functions.
4. An ability to define the following combinational circuits: multiplexer, de-multiplexers encoders/decoders, comparators, arithmetic-logic units; and to be able to build simple circuits.
5. 5. An ability to understand asynchronous and synchronous sequential circuits, like counters and shift registers.
6. An ability to understand memories like RAM and ROM, Programmable Logic Array and Programmable Array Logic.

**Syllabus:**

1. **Binary Systems:** Digital Systems. Binary Numbers. Number Base Conversions. Octal and Hexadecimal Numbers. Complements. Signed Binary Numbers. Binary Codes. Binary Storage and Registers. Binary Logic

2. **Boolean Algebra and Logic Gates:** Basic Definitions. Axiomatic Definition of Boolean Algebra. Basic Theorems and Properties of Boolean Algebra. Boolean Functions. Canonical and Standard Forms. OtherLogicOperations.Digital Logic Gates. Integrated Circuits.

3. **Combinational Logic Design, Gate-Level Minimization:** The Map Method. Four-Variable Map. Five-Variable Map. Product of Sums Simplification. Don't-Care Conditions. NAND and NOR Implementation. Other Two- Level Implementations. Exclusive-OR Function. HardwareDescriptionLanguage(HDL).

4. **Combinational Logic:** Combinational Circuits. Analysis Procedure. Design Procedure. Binary Adder- Subtractor.DecimalAdder.BinaryMultiplier.MagnitudeComparator.Decoders. Encoders. Multiplexers. HDL For Combinational Circuits.

5. **Sequential Logic Design, Synchronous Sequential Logic**: Sequential Circuits .Latches .Flip-Flops. Analysis of Clocked Sequential Circuits. HDLForSequentialCircuits.StateReductionandAssignment.DesignProcedure.

6. **Registers ad Counters:** Registers. Shift Registers. Ripple Counters. Synchronous Counters. Other Counters. HDL for Registers and Counters.

7. **Memory and Programmable Logic**: Introduction. Random-Access Memory. Memory Decoding, Error Detection and Correction. Read-Only Memory. Programmable Logic Array.

Programmable Array Logic. Sequential Programmable Devices.

**TEXTBOOK**:

1. Digital Design, 3rd Edition, M.Morris Mano, Pearson Education.

**REFERENCEBOOKS:**


1. Digital Logic Design Principles, Norman Balabanian & Bradley Carlson, John Wiley & Sons(Asia) Pvt.Ltd.,2002
2. Fundamentals of Digital Logic with VHDL Design, Stephen Brown and Zvonko Vranesic, Tata McGraw-Hill Edition,2002

| CSE 2.1.7 | DATA STRUCTURES LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** | **Univ. Exam: 3 Hours** | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To implement stacks and queues using arrays and linked lists.
2. To develop programs for searching and sorting algorithms.
3. To write programs using concepts of various trees.
4. To implement programs using graphs.

**Course Outcomes:**

1. Student will be able to write programs to implement stacks and queues.
2. Ability to implement various searching and sorting techniques.
3. Ability to implement programs using trees and graphs.

**List of Programs:**

1. Write a C program for sorting a list using Bubble sort and then apply binary search.
2. Write a C program to implement the operations on stacks.
3. Write a C program to implement the operations on circular queues.
4. Write a C program for evaluating a given postfix expression using stack.
5. Write a C program for converting a given infix expression to postfix form using stack.
6. Write a C program for implementing the operations of a dequeue
7. Write a C program for the representation of polynomials using circular linked list and for the addition of two such polynomials
8. Write a C program for quick sort
9. Write a C program for Merge sort.
10. Write a C program for Heap sort
11. Write a C program to create a binary search tree and for implementing the in order, preorder, post order traversal using recursion
12. a)Write a C program for finding the transitive closure of a digraph
    b)Write a C program for finding the shortest path from a given source to any vertex in a digraph using Dijkstra's algorithm
13. a)Write a C program for finding the Depth First Search of a graph.
    b)Write a C program for finding the Breadth First Search of a graph.

| CSE 2.1.8 | OBJECT ORIENTED PROGRAMMING LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** | **Univ. Exam: 3 Hours** | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To develop programs using basic OOPS concepts such as classes and objects.
2. To implement programs using Inheritance concepts.
3. To implement programs using Exception handling.
4. To develop programs using operator overloading concepts.

**Course Outcomes:**

1. Student will be able to use OOPs concepts.
2. Ability to apply Inheritance concepts to several problems.
3. Ability to use Exception Handling concepts.

**List of Programs:**

1. Write a Program in C++ that implements stack operations using classes and objects.
2. Write a Program in C++ performing complex number addition using friend functions.
3. Write a Program in C++ for complex number addition using operator overloading.
4. Write a Program in C++ to perform string operations by overloading operators.
5. Write a Program in C++ on hierarchical inheritance showing public, private and protected inheritances.
6. Write a Program in C++ for computation of student's result using hybrid inheritance.
7. Write a Program in C++ implementing bubble-sort using templates.
8. Write a Program in C++ on virtual functions.
9. Write a Program in C++ for handling PushOnFull and PopOnEmpty Exceptions for a Stack.
10. Write a Program in C++ for copying one file to another file using streams.
11. Write a Program in C++ for writing and reading a class object to a file.
12. Write program in C++ to implement
    a) One catch block and all Exceptions
    b) using Multiple Catch blocks.
13. Write a program in C++ to implement the finally block.
14. Write a program in C++ to implement pointers to a derived class and virtual base classes.
15. Write a program tin C++ to implement conversion of objects between different

classes using conversion functions.

16. Write a program in C++ to implement function overloading- with various data types, with different number of arguments.
17. Write a program in C++ to evaluate mixed mode expressions and implicit type conversions.
18. Write a program in C++ to show that there is ambiguity in Multiple Inheritance.
19. Write a program in C++ to implement a virtual destructor.
20. Write a program in C++ to mimic a bank management system (user logins, requests for withdraw/credit, system verifies whether enough balance is available, update the account summary, etc.)

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A) VISAKHAPATNAM**

**II – SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION**

**Branch: COMPUTER SCIENCE AND ENGINEERING**

**II/IV B.TECH** (FOUR YEAR COURSE) &
**II/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)
(With effect from **2015-2016** admitted batch onwards)
Under Choice based Credit System

| **B.TECH. (CSE) 2nd YEAR II-SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION WITH EFFECT FROM 2015-2016 BATCHES** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **SUB.REF. NO.** | **NAME OF THE SUBJECT** | **PERIODS** | | | **MAXIMUM MARKS** | | | **CREDITS** |
| | | THEORY | TUTORIAL | LAB | EXAM | SESSIONALS | TOTAL | |
| CSE 2.2.1 | OPERATING SYSTEMS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.2.2 | COMPUTER ORGANIZATION | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.2.3 | MICROPROCESSORS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.2.4 | DATA COMMUNICATIONS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.2.5 | ADVANCED DATA STRUCTURES | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.2.6 | OPERATIONS RESEARCH | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 2.2.7 | ENVIRONMENTAL STUDIES | 3 | 1 | | 70 | 30 | 100 | 2 |
| CSE 2.2.8 | OPERATING SYSTEMS LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 2.2.9 | DIGITAL ELECTRONICS & MICROPROCESSORS LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| **TOTAL CREDITS** | | | | | | | | 30 |

| CSE 2.2.1 | OPERATING SYSTEMS | |
|---|---|---|
| Instruction: 3 Periods + 1 Tut/week, Univ. Exam: 3 Hours | | Credits: 4 |
| Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks |

**Course objectives:**

1. To understand evolution of Operating System.
2. To understand operating system as a layer of abstraction above physical hardware that facilitates usage convenience and efficient resource management of computer system resources.
3. To learn design and implementation of policies and mechanisms for OS subsystem.
4. To investigate case studies to understand the design philosophies / paradigm for popular multiuser or single user operating system.

**Course Outcomes:**

1. The student understands OS evolution, its structure and services provided by it.
2. Learn process life cycle, process scheduling objectives, policies and mechanisms, process synchronization, inter process communication, deadlocks and other process subsystem related concepts.
3. Learn memory hierarchy, allocation and deallocation policies and mechanism for main and auxiliary memory, file system design and implementation issues.
4. investigate UNIX/ LINUX and Windows OS platforms w.r.t similarities and differences in design philosophies.

**Syllabus:**

1. **Introduction to Operating Systems**: Over View of Operating Systems, Types of Operating Systems, Operating System Structures, Operating System Services, System Calls, Virtual Machines, Operating System Design and Implementation.

2. **Process Management**: Process Concepts, Operations on Processes, Cooperating Processes, Threads, Inter Process Communication, Process Scheduling, Scheduling Algorithms, Multiple - Processor Scheduling, Thread Scheduling.

3. **Process Synchronization:** The Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Critical Regions, Monitors.

4. **Deadlocks**: System Model, Deadlock Characterization, Methods For Handling Deadlocks, Deadlock Prevention, Avoidance, Deadlock Detection, Recovery from Deadlocks

5. **Memory Management:** Logical versus Physical Address, Swapping, contiguous memory allocation, paging, structure of the page table , segmentation, Virtual Memory,

Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped files

6. **File Systems, Implementation, and Secondary-storage Structure:** Concept of a file, Access Methods, Directory Structure, Protection, File System Structure, Allocation Methods, Free Space Management, Directory Management, Device Drivers, overview of Mass-storage structure, Disk structure, disk attachment, disk scheduling, swap-space management.

7. **Case study:** Overview of LINUX, Windows Operating systems

**Text Book:**

1. Operating Systems, Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, John Wiley Publ., Seventh Edition.

**Reference Books:**

1. Modern Operating Systems, Andrew S. Tanenbaum, , 2nd edition, 1995, PHI.
2. Operating Systems, William Stallings 5th Edition - PHI
3. Operating Systems: A Design-Oriented Approach', Charles Crowley, 'Tata Hill Co.,1998 edition.

| CSE 2.2.2 | COMPUTER ORGANIZATION | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week, Univ. Exam: 3 Hours** | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To study about structure and functional components of a computer.
2. Understanding the hierarchical organization of a computer system which consists of instruction set of commands.
3. Learn about the architecture of a computer from a programming view.
4. To design a balance system that minimizes performance and utilization of all elements.

**Course Outcomes:**

1. Knowledge about major components of a computer such as processor, memory and I/O modules along with their interconnections internally with outside world.
2. Detailed idea about architecture of central processing unit, functions of control unit, memory, I/O devices and their issues.
3. simple and multiple processor organization and their issues.

**Syllabus:**

1. **Register Transfer and Micro operations:** Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro operations, Shift Micro operations, Arithmetic Logic Shift Unit.

2. **Basic Computer Organization and Design:** Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input- Output and Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic.

3. **Micro programmed Control**: Control Memory, Address Sequencing, Micro program Example, Design of Control Unit.

4. **Central Processing Unit:** Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer(RISC)

5. **Pipeline and Vector Processing:** Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISK Pipeline, Vector Processing, Array Processors.

6. **Input/output Organization:** Peripheral Devices, I/O interface, Asynchronous data transfer, Modes of transfer, priority Interrupt, Direct memory access, Input-Output Processor (IOP), Serial Communication.

7. **Memory Organization**: Memory Hierarchy, Main memory, Auxiliary memory, Associate Memory, Cache Memory, and Virtual memory, Memory Management Hardware.

**Text Books:**


1. Computer System Architecture, M. Morris Mano, Prentice Hall of India Pvt. Ltd., Third Edition, Sept. 2008.


**Reference Books:**


1. Computer Architecture and Organization, William Stallings, PHI Pvt. Ltd., Eastern Economy Edition, Sixth Edition, 2003.
2. Computer Organization and Architecture, Linda Null, Julia Lobur, Narosa Publications ISBN 81- 7319-609-5
3. Computer System Architecture", John. P. Hayes.

| CSE 2.2.3 | MICROPROCESSORS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** | **Univ. Exam: 3 Hours** | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To discuss the architectures of 8085, 8086 microprocessors, their instruction sets and related ALP programs.
2. To discuss interfacing semiconductor memories, interfacing peripheral to Intel 8086.
3. To study interfacing data converters to 8086 and discuss about micro controller 8051 architecture.

**Course Outcomes:**

1. Understand the basic architectures of 8085 and 8086 microprocessors.
2. Ability to write ALP programs using instruction sets.
3. Understand the various interfacing concepts and micro controllers.

**Syllabus:**

1. **Introduction to Microprocessors and Microcomputers:** A Brief Architecture and Programming of 8085 Microprocessor.
2. **Architecture**: Instruction Set and Programming of 8086 Microprocessor
3. **Interfacing Semiconductor Memories and I/O Devices**: Semiconductor Memories: Classification Internal Organization& Functional Description, Interfacing SRAMs and EPROMs to 8086, Interfacing Characteristics of I/Devices, I/O Device addressing methods, I/O Device Programming Methods.
4. **Interfacing Peripherals to Intel8086 -1:** Parallel I/O Interface- 8255,Serial I/O Interface – 8251, Timer Interface - 8253/8254
5. **Interfacing Peripheral to Intel8086 -2:**Keyboard/Display Interface- 8279,InterruptControllerInterface–8259
6. **Interfacing Data Converters to 8086:** D/A Conversion Methods, A/D Conversion methods, Interfacing DAC, Interfacing ADC.
7. **Introduction to Micro controllers:** Intel 8051Architecture and Programming

**Text Books:**


1. Microprocessor Architecture, Programming, and Applications with the 8085 Ramesh S. Gaonkar, 4th Edition, PenramInternational, 1999
2. The 80x86 Family, Design, Programming and Interfacing, John E.Uffenbeck, 3rd Edition, Pearson Education Inc., 2002
3. KennethJ.Ayala, 8051 Microcontroller Architecture, Programming And Applications,2nd Edition, Penram International Publications,1999

**Reference Books:**


1. BARRY B. BREY, The Intel Microprocessors 8086/8088,80186/80188,80286,80386 and 80486, Pentium, PentiumPro Processor, Pentium II, Pentium III, Pentium 4, Architecture, Programming and Interfacing, 8th Edition, Pearson Education Inc.,2009

2. Walter A. Tribel and Avtar Singh, The8088 and 8086 Microprocessors, Programming, interfacing, Software, Hardware, and Applications, 4th Edition, Pearson Education Inc., 2003 3.MicroprocessorsandInterfacing, Programming and Hardware, 2ndEdition, Douglass V. Hall, TMH Edition, 1999

3. Sanjay K Bose, Hardware and Software of Personal Computers, New Age International (P) Ltd., 1991
4. Myke Predko, Programming and Customizing the 8051 Microcontroller, TMH, 1999

| CSE 2.2.4 | DATA COMMUNICATIONS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week, Univ. Exam: 3 Hours** | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To study basics of data communication systems.
2. To study the various types of transmission media.
3. To study the various hardware concepts related to data communications.
4. To discuss about modem and multiplexing techniques.

**Course Outcomes:**

1. Student will able to understand basic concepts related communication systems.
2. Ability to understand different transmission medias
3. Ability to understand concepts related to data communication hardware.
4. Ability to understand basic functionality of modems.

**Syllabus:**

1. **Introduction to Data Communications:** A Communications Model, Data Communications and Data Communications Networking, Protocols and Protocol Architecture, Characteristics of Data Transmission: Concepts and Terminology, Analog and Digital Data Transmission, Transmission Impairments

2. **Transmission Media:** Guided Transmission Media, Wireless Transmission Data Encoding, Digital Data, Digital Signals, Digital Data, Analog Signals, Analog Data, Digital Signals, Analog Data, Analog Signals

3. **Data Communication Interface:** Asynchronous and Synchronous Transmission, Line Configurations, Interfacing. Data Link Control Flow Control, Error Detection, Error Control, High-Level Data Link Control (HDLC),Other Data Link Control Protocols.

4. **Data Communications Hardware:** Terminals : Introduction, Basic Terminal Components, Enhanced Terminal Components, General-Purpose Terminals, Remote Job Entry Terminals, Transaction Terminals, Clustering of Terminal Devices.

5. **Communications Processing Hardware:**
   Introduction,SwitchingProcessors,MultidropLines,Multiplexers,Concentrators, Front-End Processors.

6. **Modems:** Network Attachment and Regulations, Line Conditioning and Leased Lines, Modems and Modem Circuits.

7. **Multiplexing:** Frequency-Division Multiplexing, Synchronous T ime-Division Multiplexing: Characteristics, TDM Link Control, Digital Carrier Systems Statistical Time-Division Multiplexing: Characteristics.

**TEXTBOOKS:**


1. William Stallings, Data and Computer Communications, 10th Edition, PH/Pearson Edu. Inc., 2014
2. Mary E.S. Loomis, Data Communications, PHI-N.J.,1983(Chapter 3, Chapter 5)
3. PaulBates,PracticalDigitalandDataCommunications,PHI-N.J,1987(Chapter5)


**REFERENCEBOOKS:**


1. Behrouz A. Forouzan, Data Communications and Networking, 3rdEditionTMH,2004
2. William A. Shay, Understanding Data Communications & Networks, 2ndEdition Thomson-Brooks/Cole –Vikas Publishing House,1999
3. Michale A. Miller, Data & Network Communications, Thomson/Delmar –Vikas Pub. House, 2000

| CSE 2.2.5 | ADVANCED DATA STRUCTURES | |
|---|---|---|
| Instruction: 3 Periods + 1 Tut/week, Univ. Exam: 3 Hours | | Credits: 4 |
| Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks |

**Course Objectives:**

1. To study the concepts related to trees such as binary trees, BST, AVL trees etc.
2. To discuss various hashing technique.
3. To study the various external sorting algorithms.
4. To discuss the concepts related to disjoint set ADT.
5. To study several graph algorithms and their time complexities.

**Course outcomes:**

1. Student will be able to write programs to implement various trees.
2. Ability to understand various hashing techniques.
3. Ability to write programs to implement sorting techniques.
4. Ability to understand concepts related to graph theory.

**Syllabus:**

**1. Trees:** Definition , operations and applications of Binary search trees, AVL trees, Red-Black Trees, Splay trees, Tries and B-Trees, B+ Trees

2. **Hashing:** Hash Table Structure, Hash Function, Collision handling, Separate Chaining, Open Addressing, Rehashing, Extendible hashing

**3. Priority Queues:** Heap model and implementations, Binary Heap, Applications of Priority Queues, d-Heaps, Leftist Heaps, Skew Heaps, Binomial Queues structure, operations and implementation

**4. External sorting:** Difference between internal and external sorting, Model and simple algorithm for External sorting, Multi-way Merge, Poly-phase Merge, Replacement selection

**5. Disjoint Set ADT:** Equivalence relations, Dynamic equivalence problem, Basic data structure, smart union algorithms, path compression, Analysis of union/find algorithm, applications of ADT Disjoint set

**6. Graph algorithms:** Representation of graphs, Topological sort, Network flow problems, Applications of Depth first search for finding Bi-connectivity, Euler circuits, strong components, Introduction of NP-Completeness

**7. Amortized analysis:** Introduction to amortized analysis, Basic approaches, binary queues, Fibonacci heaps ,skew heaps and splay trees.

**Text Book:**

1. Data Structures and Algorithm Analysis in C – Mark Allen Weiss, Pearson Edu Publishers.

2. Data Structures and Algorithms: Concepts, Techniques and Applications – G.A.V.Pai, Tata Mc Graw Hill Publishers

**References:**

1. Advanced Data Structures – Peter Brass, Cambridge University Press, 2008

| CSE 2.2.6 | OPERATIONS RESEARCH | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week, Univ. Exam: 3 Hours** | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives**

1. To discuss about basic Operation Research concepts , Formulation of LPP and its solution using graphical method.
2. To discuss about standard form of LPP. solving LPP using various methods.
3. To study the various solutions of transportation problems and assignment problems.
4. To discuss about PERT and CPM charts
5. To discuss about replacement problems, inventory problems and game theory.

**Course Outcomes:**

1. Ability to solve LPP problems using various methods.
2. Ability to solve transportation and assignment problems using several methods.
3. Analyze the PERT and CPM charts
4. Ability to solve replacement problems and game theory problems.

**Syllabus:**

1. Overview of Operations Research, Types of OR Models , Phases of Operations Research– OR Techniques, Introduction to Linear Programming, Formulation of Linear Programming Problem, Graphical Solution; Graphical Sensitivity Analysis,

2. Standard Form of LPP, Basic Feasible Solutions , Unrestricted Variables, Simplex Algorithm , Artificial Variables, Big M Me th od , Two Phase Simplex Method, Degeneracy, Alternative Optimal, Unbounded Solutions, Infeasible Solutions, Primal And Dual Problems And Their Relations, Dual Simplex Method

3. Transportation Problem as LPP, Initial Solutions, North West Corner Rule, Lowest Cost Method, Vogels Approximation Method, Optimum Solutions of TPP, Degeneracy in Transportation, Transportation Algorithms ,

4. Assignment Problem , Assignment Problem as LPP, Hungarian Method, Travelling Salesman Problem, Solutions Of TSP, Sequencing Problems, N-Jobs Two Machine Problems, N-Jobs K Machines Problems, Two-Jobs M- Machine Problems, Crew Scheduling Problems

5. Network Representation of A Project, CPM and PERT , Critical Path Calculations, Time – Cost Optimizations, PERT Analysis and Probability Considerations, Resource Analysis in Network Scheduling.

6. Replacement Problems-Individual And Group Replacement Policy, Reliability & System Failure Problems, Inventory-Factors Effecting Inventory-EOQ, Inventory Problems With

and Without Shortages, Inventory Problems With Price Breakups, Multi Item Deterministic Problems. Probabilistic Inventory Problems

7. Game Theory : Two Person Zero Sum Games , Mixed Strategy Games and Their Algorithms.

**Text Books:**

1. Operations Research, Kanti Swaroop, P.K. Gupta, Man Mohan, Sulthan Chand& Sons Education

2. Publishers Operations Research – An Introduction, Handy A Taha – Pearson Education .

**Reference B:**

1. Operations Research Panneer Selvan Prentice Hall Of India.
2. Operations Research By S.D Sharma
3. Introduction To Operations Research, F.S. Hiller, G.J. Liberman, TMH
4. Operations Research, Richard Bronson, Schaum's Series, Mcgrawhill

| CSE 2.2.7 | ENVIRONMENTAL STUDIES | |
|---|---|---|
| Instruction: 3 Periods + 1 Tut/week, Univ. Exam: 3 Hours | | Credits: 2 |
| Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks |

**Course Objectives:**

The Program seeks to provide students better understanding and planning for conservation through an interdisciplinary environmental science curriculum that is designed to enhance scientific inquiry and to strengthen scientific competence. Through these efforts, the Program aims at preparing and providing students to opportunities for careers in environmental sciences, environmental health, public health, and medical schools.

**Course Outcomes :**

1. Recognize major concepts in environmental sciences and demonstrate in-depth understanding of the environment.
2. Develop analytical skills, critical thinking, and demonstrate problem-solving skills using scientific techniques.
3. Demonstrate the knowledge and training for entering graduate or professional schools, or the job market

**Module 1: Introduction (1 lecture)**

(a) Definition, Scope and importance
(b) Measuring and defining environmental development: indicators

**Module 2: Ecosystem (2 lectures)**

(a) Introduction, types, characteristic features, structure and functions of Ecosystems -Forest –Grass land -Desert -Aquatic (lakes, rivers and estuaries)

**Module 3: Environmental and Natural Resources management (8 lectures)**

a) Land resource-Land as a resource -Common property resource -Land degradation -Soil erosion and desertification -Effects of modern agriculture, fertilizer – pesticide problems
b) Forest resources Use and over-exploitation-Mining and dams- their effects on forest and tribal people
c) Water resources-Use and over-utilization of surface and ground water-Floods and droughts-Water logging and salinity-Dams –benefits and costs-Conflicts over water
d) Energy resources
e) Energy needs-Renewable and non-renewable energy source-Use of alternate energy sources -Impact of energy use on environment

**Module 4: Bio-diversity and its conservation (3 lectures)**

a) Value of bio-diversity-consumptive and productive use, social, ethical, aesthetic and option values
b) Bio-geographical classification of India- India as a mega diversity habitat
c) Threats to biodiversity- Hot spots, habitat loss, poaching of wildlife, loss of species, seeds etc.
d) Conservation of bio-diversity- In-situ and Ex-situ conservation

**Module 5: Environmental Pollution Local and Global Issues (8 lectures)**

a) Cause, effects and control measures of Air Pollution- Indoor air pollution-Water pollution- Soil pollution- Marine pollution-Noise pollution-Solid waste management, composting, vermiculture- Urban and industrial wastes, recycling and reuse
b) Nature of thermal pollution and nuclear hazards
c) Global Warming
d) Acid rain
e) Ozone depletion

**Module 6 : Environmental problems in India (5 lectures)**

a) Drinking water, Sanitation and Public health
b) Effects of activities on the quality of environment, Urbanization-Transportation-Industrialization-
c) Green revolution
d) Water scarcity and Ground Water depletion
e) Controversies on major dams- resettlement and rehabilitation of people: problems and concerns
f) Rain water harvesting, cloud seeding and watershed management

**Module 7: Economy and Environment (4 lectures)**

a) The economy and environment interaction
b) Economics of development, preservation and conservation
c) Sustainability: theory and practice
d) Limits to Growth
e) Equitable use of resources for sustainable lifestyles
f) Environmental Impact Assessment

**Module 8: Social Issues and the Environment (2 lectures)**

a) Population growth and environment
b) Environmental education
c) Environmental movements
d) Environment vs Development

**Module 9: Institutions and Governance(5 lectures)**

a) Regulation by Government
b) Monitoring and Enforcement of Environmental regulation
c) Environmental Acts Water (Prevention and Control of pollution) act-Air (Prevention and Control of pollution) act-Envt. Protection act-Wild life Protection act-Forest Conservation act-Coastal Zone Regulations
d) Institutions and policies relating to India
e) Environmental Governance

**Module 10: International Conventions( 2 lectures**)

a) Stockholm Conference 1972
b) Earth Summit 1992
c) World Commission for environmental Development (WCED)

**Module 11: case Studies ( 3 lectures)**

a) Chipko movement
b) Narmada Bachao Andolan
c) Silent Valley Project
d) Madhura Refinery and Taj Mahal
e) Industrialization of Pattancheru
f) Nuclear reactor in Nagarjunasagar
g) Tehri dam
h) Ralegaon Siddhi (Anna Hazzare)
i) Kolleru lake-aquaculture
j) Florosis in Andhra Pradesh

**Module 12: Field Work( 5 lectures**)

a) Visit to a local area to document and mapping environmental assests- river/ forest/ grassland/ Hill/ Mountain.
b) Study of local environment- common plants, insects, birds
c) Study of simple ecosystems- pond, river, hill, slopes etc.
d) Visit to Industries, Water treatment plants, affluent treatment plants

| CSE 2.2.8 | OPERATING SYSTEMS LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** Univ. Exam: 3 Hours | | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To learn about UNIX/LINUX operating system, its intervals.
2. To learn system programming for UNIX/LINUX Operating System.
3. To understand UNIX/LINUX shell and its programming.
4. To understand resource management policies and mechanisms and their performance evaluation.

**Course Outcomes:**

1. The student practices UNIX commands, Vi editor, shell commands.
2. The student develops skill in writing C programs using system calls for process management, inter process communication and other aspects.
3. The student learns shell programming and develops skill for writing scripts for batch level tasks.
4. The student learns to simulate OS resource management aspects like process scheduling , page replacement and others to evaluate performance.

**Module I**

1. OS lab familiarization, Home Assignment on Unix commands, Vi editor
2. Simple C programs using command line arguments, system calls, library function calls, make utility
3. C programs using fork system call to create processes and study parent, child process mechanism
4. C programs to create process chaining, spawning
5. C programs to handle errors using errno, perror() function
6. C programs to use pipe system call for inter process communication

**Module II**

1. Familiarization of Unix shell programming
2. Simple shell programming exercises
3. Shell programming using decision making constructs
4. Shell programming using loop constructs
5. Shell programming for file and directory manipulation

**Module III**

1. C programs to study process scheduling implementing FCFS, Shortest Job First, and Round Robin algorithms
2. C programs to study page replacement implementing FIFO, Optimal, and LRU page replacement algorithms
3. C programs to study deadlock avoidance and detection
4. C Programs to simulate free space management

**References:**


1. Unix concepts and applications by Sumitabha Das,

   TMH Publications.
2. Unix programming by Stevens, Pearson Education.
3. Shell programming by YashwanthKanetkar.
4. Operating SystemConcepts bySilberschatz, and Peter Galvin.

| CSE 2.2.9 | DIGITAL ELECTRONICS & MICROPROCESSORS PROGRAMMING LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** Univ. Exam: 3 Hours | | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**Course Objective:**

1. To learn the about logic gates, half adders, full adders and flip -flops.
2. To learn about the microprocessor programming.
3. To learn about the microprocessor interfacing with stepper motor, R-2R ladder.

**Course Outcomes:**

1. The student understands the logic gates, half adders, full adders and flip-flops to design a circuit.
2. The student develops the skill of writing microprocessor programming.
3. The student understands the interfacing of microprocessor with stepper motor, R-2R ladder.

**1. DIGITAL EXPERIMENTS**

Verification of truth tables of OR, AND, NOT, NAND, NOR, EX-OR gates (By using 7400-series)

Construction of gates using NAND, NOR gates.

Construction of Half and Full adders and verifying their truth tables.

Operation and verifying truth tables of flip- flops- RS, D, and JK using ICs.

Construction of Decade counters (7490).

Driving Stepper motor using JK flip-flop

Simulation experiments using appropriate electronic circuit simulation.

4-bit parallel adder using combinational circuits.

Decade counter using JK flip flops.

Up/Down counter using JK flip flop.

Up/Down counter using 7493.

**2. MICROPROCESSOR (Software)**

Binary addition & subtraction. (8-bit & 16-bit)
Multiplication & division.
Picking up largest/smallest number.
Arranging –ascending/descending order.
Decimal addition (DAA) & Subtraction.
Time delay generation

**3. MICROPROCESSOR (Hardware)**

Interfacing R-2R Ladder network (DAC) (4 bits) to generate waveforms.

Interfacing a stepper motor and rotating it clockwise/anti clockwise through a known angle.
Interfacing a seven segment display.

Interfacing ADC for temperature measurement.

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING(A) VISAKHAPATNAM**

**I– SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION**

**Branch: COMPUTER SCIENCE AND ENGINEERING**

**III/IV B.TECH** (FOUR YEAR COURSE) &

**III/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)

(With effect from **2015-2016** admitted batch onwards)

Under Choice based Credit System

**B.TECH. (CSE) 3rd YEAR I-SEMESTER SCHEME WITH EFFECT FROM 2015-2016**

| SUB.REF. | NAME OFTHE SUBJECT | PERIODS | | | MAXIMUM MARKS | | | CREDITS |
|---|---|---|---|---|---|---|---|---|
| | | THEORY | TUTORIAL | LAB | EXAM | SESSIONALS | TOTAL | |
| CSE 3.1.1 | COMPUTER NETWORKS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.1.2 | MOOCS-I | -- | -- | -- | 70 | 30 | 100 | 2 |
| CSE 3.1.3 | WEB TECHNOLOGIES | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.1.4 | FORMAL LANGUAGES & AUTOMATA THEORY | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.1.5 | DATABASE MANAGEMENT SYSTEMS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.1.6 | ELECTIVE-I | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.1.7 | MOOCS-II (Un-audit) | -- | -- | -- | -- | -- | -- | -- |
| CSE 3.1.8 | DATABASE MANAGEMENT SYSTEMS LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 3.1.9 | DATA COMMUNICATIONS AND COMPUTER NETWORKS LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 3.1.10 | SOFT SKILLS LAB | -- | -- | 3 | | 100 | 100 | 2 |
| **TOTAL CREDITS** | | | | | | | | **28** |

**ELECTIVE-I:**

1) Application Development Using Java, 2) Advanced Microprocessors, 3) Digital Signal Processing, 4) Principles Of Programming Languages, 5) Image Processing

**MOOCS :**

Each student should learn any one of the following topics by registering for courses through Online instruction from standard e-learning portals like nptel, coursera, etc. and write the examination conducted as per the university norms.

**List of topics for MOOCS-I:**

1)Data Visualization using Tableau, 2)Python Programming, 3)Mobile Application Development, 4)Modeling and simulation using MatLab, 5)Recommender Systems.

**List of topics for MOOCS-II:**

1) Bio Informatics, 2) Geo Informatics, 3) E- Commerce, 4)Semantic Web.

| CSE 3.1.1 | COMPUTER NETWORKS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To make the students understanding of basic requirements of network hardware, software and its architecture.
2. Familiarize the students with layered architecture of the network software and hierarchal nature of the network physical infrastructure.
3. Study of various network interconnecting devices and other associated network hardware.
4. Introduction of advanced networking concepts and wireless and wireless sensor networks.

**Course Outcomes:**

1. The student must be able to understand the design and estimate the requirements for practical setup of a given network scenario and size.
2. Realize the Operation, maintenance and management of the Internet by mapping the theoretical networking concepts to the real-time network scenarios.
3. Demonstrate the applications of wireless Networks and over view of advanced networking concepts.
4. Identify different networking devices and their usage and functionality

**Syllabus:**

1. **Introduction to Computer Networks:** Introduction, Network Hardware, Network Software, Reference Models, Network Examples, Internet Based Applications.

2. The Medium Access Control: The Channel Allocation Problem, CSMA Protocols, Collision Free Protocols, The Ethernet, Wireless LANS, Bluetooth

3. **Network Layer :** Network Layer Design Issues, Routing Algorithms, Congestion Control Algorithms, Net work Layer in the Internet, IP Protocol, IP Address, Subnets, and Internetworking.

4. **Transport layer:** Transport Service, Elements of Transport Protocols, TCP and UDP Protocols, Quality of Service Model, Best Effort Model, Network Performance Issues.

5. **Application Layer:** Over View of DNS, SNMP, Electronic Mail, FTP, TFTP, BOOTP, HTTP Protocols, World Wide Web, Firewalls.

6. **Network Devices:** Over View of Repeaters, Bridges, Routers, Gateways, Multiprotocol Routers, Hubs, Switches, Modems, Channel Service Unit CSU, Data Service Units DSU, NIC, Wireless Access Points, Transceivers, Firewalls, Proxies.

7. Overview of Cellular Networks, Ad-hoc Networks, Mobile Ad-hoc Networks, Sensor Networks

**Text Books:**

1. Computer Networks, Andrews S Tanenbaum,, 5th Edition, Pearson Edu. Inc., 2011

**References:**

1. Data Communications and Networking , Behrouz A Forouzan , Tata McGraw-Hill Co Ltd , Second Edition, ISBN: 0-07-049935-7
2 . Computer networks, Mayank Dave, CENGAGE.
2. Computer networks, A system Approach, 5thed, Larry L Peterson and Bruce S Davie, Elsevier.
3. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
4. Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson.

| CSE 3.1.3 | WEB TECHNOLOGIES | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

On completing this course student will be able to

1. Understand the principles of Web based application development.
2. Design dynamic content in Web Pages using JavaScript.
3. Understanding the concepts of java Servlets, java Server Pages and design applications using them.
4. Understand the concepts of Component development and design applications by establishing connections to Databases

**Course Outcomes :**

1. Students will be able to construct web based applications and Identify where data structures are appearing in them.
2. Students will be able to connect java programs to different databases.
3. Students will be able to develop EJB programs

**Syllabus:**

**1.** 1.Introduction to HTML , Core Elements , Links and Addressing, Images , Text , Colors and Background, Lists, Tables and Layouts , Frames, Forms , Cascading Style Sheets.

**2.** Introduction to Java Scripts, Elements of Objects in Java Script, Dynamic HTML with Java Script

**3.** Document type definition, XML Syntax, XML Schemas, Document Object model, Presenting XML, Using XML Processors

**4.** **4**.Introduction to Servlet, Servlet Life Cycles, Servlet Basics, Tomcat Web Server, Configuring Apache Tomcat, Handling Client Request and Response, Handling Cookies, Session Tracking.

**5.** Introduction to PHP, Language Basics, Functions, Strings, Arrays.

**6.** Web Techniques, Data bases, Graphics, PDF, Dates and Times.

**7.** MYSQL Installation, Accessing MySQL Using PHP, Form Handling, Cookies, Sessions, and Authentication,Tables, Inserting Data into Tables , Selecting Data from a Table,Updating Table , Deleting data from Table, Webpage creation.

**Text Books:**


1. Web Programming, building internet applications, 2nd Ed., Chris Bates, Wiley Dreamtech

2. The complete Reference HTML and DHTML, Thomas A. Powey

3. Learning Php, Mysql, Robin Nixon

4. Programming Php, Kevin Tatroe, Peter MacIntyre & Rasmus Lerdorf foreword by Michael Bourque.

**Reference Books:**

1. Internet , World Wide Web , How to program, Dietel , Nieto, PHI/PEA

2. Web Tehnologies, Godbole, kahate, 2nd Ed., TMH

| CSE 3.1.4 | FORMAL LANGUAGES & AUTOMATA THEORY | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course objectives:**

1. To introduce the concepts in automata theory and theory of computation to design grammars and recognizers for different formal languages.
2. To Employ finite state machines to solve problems in computing.
3. To introduce finite state machines ,context free grammars and Turing Machines and their properties as the basis for the formal expressivity of computer languages for solving linguistic decision problems.
4. To understand the concepts of tractability and decidability, the concepts of NP-completeness and NP-hard problem and also the challenges for Theoretical Computer Science and its contribution to other sciences.

**Course outcomes:**

1. Ability to think analytically and intuitively for problem-solving situations in related areas of theory in computer science
2. Ability to describe the language accepted by an automata or generated by a regular expression or a context-free grammar;
3. Ability to Understand the functioning of Finite-State Machines, Deterministic Finite-State Automata, Nondeterministic Finite-State Automata and Pushdown Automata and Turing Machines.

**Syllabus:**

1. Definitions of alphabet, strings, language, grammar, types of grammar, types of machines, generation of languages from grammar, construction of grammar from the given description of languages

2. Definition of finite state machine, Definite state machine, indefinite state machine, representations in mathematical diagram, tabular etc., id of finite state machine's, design of finite state machine from the given description, elimination of e-transitions , indefinite state machine to definite state machine, optimization of finite state machine

3. Conversion of regular grammar to finite state machine, finite state machine to regular grammar, discussion of pumping lemma, systematic way of construction of finite state machine

4. Definition of regular expression, regular algebra, minimization of regular expressions, closure properties, construction of regular expression from the given description, regular expression to finite state machine, finite state machine to regular expression, construction of regular expression for the given finite state machine- a systematic way using Arden's theorem

5. Definition of push down machine, push down machine, types of push down machine's, push down machine to context free grammar, context free grammar to push down machine, design methodology of various push down machine's, push down machine by

empty stack, push down machine by final states, conversion from one type to other type, applications of push down machine's

6. Parsing tree, bottom-up parsing, top-down parsing, types of context free grammar's, left-most and right most derivations, productions, reductions, optimization of context free grammar's, elimination of ϵ productions, unit productions, normal forms- cnf, gnf

7. Definition of Turing machine, ways of representing Turing machine's- tabular form, diagram, mathematical form, quintuples etc., design of Turing machine, id of Turing machine, types of Turing machine, halting problem, church's thesis, universal Turing machine, Gödel number, definitions of recursive functions- prf, rf, decidability.

**NOTE :** Theorem proofs are eliminated

**Text books :**

1. Introduction to automata theory, languages and computation, John.E.H.P croft/ Rajeev Motwani & JD Ullman—pearson education- III edition

**Reference Books :**

1.Theory of computation, K.L.P.Mishra and N.Chandrasekhar, PHI

2. Theory of computation, formal languages and automata theory, G P Saradhi Varma, B.Thirupathi Rao –Sci Tech publications.

| CSE 3.1.5 | DATABASE MANAGEMENT SYSTEMS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To learn the evolution of DBMS Versus File systems, data models, and layers of abstraction.
2. To understand conceptual and physical aspects of database design.
3. To learn formal and commercial query language specifications.
4. To understand concurrency control, recovery management, and other related issues.

**Course Outcomes:**

1. The student will understand ER-modeling for conceptual database design and relational model.
2. The student is introduced to formal and commercial query languages : Relational Algebra, calculus and SQL.
3. The student will learn schema refinement and normalization.
4. The Student understands locking protocols concurrency control, and crash recovery methods.

**Syllabus:**

1. **Introduction**: File system versus a DBMS , Advantages of a DBMS, Describing and Storing Data in a DBMS, The Relational model, Levels of abstraction, Data Independence, Transaction management, Structure of a DBMS.

2. **Introduction to Database Design and The Relational Model**: Database Design and ER Diagrams, Entities, Attributes and Entity Sets, Relationships & Relationship Sets, Additional Features of the ER Model, Conceptual Design with ER Model, Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Querying Relational Data, Logical Database Design: ER to Relational, Introduction to Views, Destroying/ Altering Tables and Views.

3. **Relational Algebra and SQL**: Preliminaries, Relational Algebra, The form of a Basic SQL Query, UNION, INTERSECT and EXCEPT, Nested Queries, Aggregate Operators, Null Values, Complex Integrity Constraints in SQL, Triggers and Active Databases, Embedded SQL, Dynamic SQL, JDBC.

4. **Database Design**: Schema Refinement and Normal Forms, Introduction to Schema Refinement, Functional Dependencies, Reasoning about FD's, Normal Forms, Properties of Decomposition, Normalization, Other kinds of Dependencies.

5. **Transaction Management**: The ACID Properties, Transactions & Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control.
6. **Concurrency Control**: 2PL, Serializability and Recoverability, Introduction to Lock Management, Lock Conversions, Dealing with Deadlocks, Specialized Locking

Techniques, Concurrency Control without Locking.

7. **Crash Recovery**: Introduction to ARIES, The Log, Other Recovery-Related Structures, The Write-Ahead Log Protocol, Check pointing, Recovering from a System Crash, Media Recovery.

**Text Book:**

1. Database Management Systems; Raghu Ramakrishnan, Johannes Gehrke 4th Edition, McGraw- Hill

**Reference:**

1. Database System Concepts; A. Silberschatz, H. Korth 5th Edition, McGraw-Hill

| CSE 3.1.6 | ELECTIVE-I APPLICATION DEVELOPMENT USING JAVA | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. Study of object oriented programming.
2. Learn about web based applications such as AWT components.
3. Study of multitasking by using multithreading concept.
4. Learn about network programming and applications development.

**Course Outcomes:**

1. Development of projects for web based and internet applications.
2. Exposure of network programming.
3. Idea about multitasking and multiprogramming development

**Syallabus:**

1. **Overview of Java,** Java Versions and Application Areas, **Basic Java Syntax,** Accessing arrays, Looping, Using if statements, Comparing strings, Building arrays.

2. **Basic Object-Oriented Programming in Java,** Instance variables (data members, fields),Methods (member functions),Constructors, Overloading, Encapsulation and accessor methods, JavaDoc, Inheritance, Abstract classes, Interfaces, @Override, The class path, Packages, Visibility modifiers (public, private, protected, default), JavaDoc options.

3. **3.Applets and Basic Graphics,** Applet restrictions, ,The applet life-cycle and the idea of life-cycle methods in general, Methods available for drawing operations, Loading and drawing images, Using try/catch blocks, Controlling image loading

4. **Basic File IO with the NIO Package,** Simple file reading: all lines at once into List, Simple file writing: all at once from a List, Some simple file reading and writing utilities, Faster and more flexible file reading

5. **AWT Components,** Basic AWT windows, Canvas, Panel, Frame, Processing events in GUI controls, Basic AWT user interface controls, Button, checkbox, radio button, list box Event-handling options , Handling events with separate listeners, Handling events by implementing interfaces, Organizing Windows with Layout Managers**,** Standard layout managers, Flow Layout, Border Layout, Card Layout, Grid Layout, GridBagLayout,

6. **Multithreaded Programming,** Why threads?, Three variations on the theme, Separate classes that implement Runnable, Main app implements Runnable, Inner classes that implement Runnable, Race conditions and synchronization

7. **Network Programming: Clients,** Creating sockets, Implementing a generic network client, Parsing data: StringTokenizer ,Getting user info from a mail server, Retrieving files from an HTTP server, Retrieving Web documents by using the URL class, Network

Programming: Servers**,** Steps for creating a server, Create a Server Socket object, Create a Socket object from ServerSocket,Create an input stream, Create an output stream, A generic network server, Single threaded , Multithreaded.

**TEXT BOOK**

1. Timothy Budd, "Understanding Object-oriented programming with Java", Updated Edition, Pearson Education, 2000.

**REFERENCE**

1. C. Thomas Wu, "An introduction to Object-oriented programming with Java", Fourth Edition, Tata McGraw-Hill Publishing company Ltd., 2006.

| CSE 3.1.6 | ELECTIVE-I ADVANCED MICROPROCESSORS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives :**

1. To learn about the architecture of microprocessor 32-bit, 64-bit , single core etc..
2. To learn about the microprocessor programming using Assembly language with C/C++, modular programming etc.
3. To learn about the architecture of x86 and Pentium processor architecture.
4. To learn about the architecture of microcontroller.

**Course Outcomes:**

1. The student understand the architecture microprocessor 32-bit, 64-bit , single core etc..
2. The student understand microprocessor programming using Assembly language with C/C++, modular programming etc.
3. The student understand the architecture of x86 and Pentium processor architecture.
4. The student understand the architecture of microcontroller.

**Syllabus**

1. **Advanced Microprocessor Architecture: General Structure of Microprocessors:** 32 bit and 64 bit single core, Dual core and Quad core, Internal Microprocessor Architecture, Real Mode Memory Addressing, Protected Mode Memory addressing, Memory Paging, Data addressing Modes, Program Memory Addressing Modes, Stack Memory Addressing Mode.

2. Data Movement Instructions, Program Control Instructions, Arithmetic and Logic Instructions

3. **Microprocessor Programming**: Modular Programming, Using Keyboard and Video Display , Data Conversions, Disk Files, Interrupt hooks, Using Assembly Languages with C/C++ for 32 Applications

4. **x86 and Pentium Processors Architecture:** Block Diagram, Signal groups, Memory Organization of i386 and i486 Microprocessors; Pentium Microprocessor-Special Pentium Registers, Pentium Memory Management, Pentium Processor, Pentium 4 Processors

5. **Reduced Instruction set Principles:** RISC Versus CISC, RISC Properties, RISC Evaluation, On Chip Register file versus Cache Evaluation, , Over view of RISC Development and Current systems

6. **Advanced RISC Microprocessors:** Introduction, Accessing External Memory in RISC Systems, Reducing Branch Penalties, Branch Prediction, ARM Processors, ARM Registers, ARM Instructions, ARM Built-in- shift Mechanism, ARM Branch Instruction,

Sequence Control, Data Movement and Memory Reference Instructions, Sun SPARC RISC Microprocessors and its Architecture

7. **Microcontrollers**: Introduction to 16-Bit and 32- Bit Micro Controller, 8096/8097 Architecture, CPU Registers, RALU, , Internal Program and Data Memory Timers**,** High Speed Input and Output, Serial Interface, , I/O Ports, Interrupts, Instruction Set, External Memory Interfacing, External I/O Interfacing.

**TEXT BOOKS:**


1. The Intel Microprocessors, Barry B. BREY, Sixth Ed., Pearson Education, 2003
2. Advanced Microprocessor, Danial Tabak, McGraw Hill, Inc. 2nd Ed, 2005


**REFERENCEBOOKS:**


1. Advanced Microprocessor, A.K. Ray, Tata Mc Graw Hill, 2006
2. Intel Corporation Data sheets ( www.intel.com)
3. John Peatmen, Design with Micro Controller, Mc Graw Hill, Inc.
4. Alan Clements, Principles of Computer Hardware, Oxford University Press 3rd Edition 2003

| CSE 3.1.6 | ELECTIVE-I **DIGITAL SIGNAL PROCESSING** | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To study the role of signals and systems in engineering.
2. To discuss filtering methods based on DFT and FFT.
3. To study different design procedures of IIR filters.
4. To discuss FIR filters using windowing techniques.
5. To discuss applications of digital signal processing.

**Course Outcomes:**

1. Student will be understand the role of signals and systems in engineering.
2. To design filtering methods based on DFT and FFT.
3. To describe different design procedures of IIR filters.
4. To design FIR filters using windowing techniques.
5. To identify applications of DSP.

**Syllabus:**

1. **Introduction :** Signals Systems and Signal Processing, classification of signals, analog to digital and digital to analog signals conversion fundamentals. Time-domain representation of continuous-time signals, Frequency-domain representation of continuous-time signals: Fourier series, Fourier Transform and their properties; Sampling theorem.

2. **DSP and Applications :** Introduction to Digital Signal processing (DSP) and its applications, Discrete- time signals, Discrete-time systems, Linear time invariant systems and their properties, Linear constant – coefficient difference equations, Frequency domain representation of discrete-time signals and systems.

3. **Correlation of discrete time signals :** Cross correlation and auto correlation sequences, properties of autocorrelation and cross correlation sequences, correlation of periodic sequences, computation of correlation sequences, input output correlation sequences.

4. **Frequency-domain representation of discrete-time signals :** The discrete Fourier series (DFS) and the discrete Fourier transform (DFT), properties of the DFT, Efficient computation of DFT: Radix-2 FFT algorithms.

5. **Z Transforms :** The z-Transform, properties of the z-Transform, inverse z-Transform, analysis of linear time invariant systems in z-domain, One-sided z-Transform: Definition and properties, Solution of Difference equations. Realization of FIR and IIR

system: Direct Form-I, Direct Form-II, Cascade and Parallel realizations.

6. **Design of IIR Filters :** Introduction, IIR filter design using Butterworth and Chebyshevapproximations, Impulse invariant transformation, Step invariant transformation, Bilinear transformation; Frequency- domain transformations.

7. **Design of FIR Filters :** Introduction, linear-phase FIR filters, FIR filter design: Window method, Frequency Sampling Method, Difference between IIR and FIR filters.

**Text Books:**


1. Digital Signal Processing, Principles, Algorithms, and Applications: John G. Proakis, Dimitris G. Manolakis, Pearson Education / PHI, 2007.
2. Discrete Time Signal Processing – A.V.Oppenheim and R.W. Schaffer, PHI


**Reference Books:**


1. Digital Signal Processing: MH Hayes, Schaum's Outlines, TATA Mc-Graw Hill, 2007.
2. DSP Primer - C. Britton Rorabaugh, Tata McGraw Hill, 2005.
3. Fundamentals of Digital Signal Processing using Matlab – Robert J. Schilling, Sandra L. Harris, Thomson, 2007.

| CSE 3.1.6 | ELECTIVE-I | PRINCIPLES OF PROGRAMMING LANGUAGES |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course objectives:**

1. To learn the underlying principles and concepts of programming language.
2. To understand programming language translation process.
3. To expose students to the important paradigms of programming.
4. To understand the concepts of distributed processing and network programming.

**Course outcomes:**

1. Ability to compare different programming languages.
2. Ability to discuss the significant achievements in programming language history.
3. Ability to assess the programming languages in scientific manner.

**Syllabus:**

1. **Language Design Issues**: Study Programming Languages, History of Programming Languages, Role of Programming Languages, Programming Environments.

2. **Impact of Machine Architectures**: Operation of a Computer, Virtual Computers and Binding Times**;** Language Translation Issues: Programming Language Syntax, Stages in Translation, Formal Translation Models, Recursive Descent Parsing; Modeling Language Properties: Formal Properties of Languages, Language Semantics.

3. **Elementary Data Types**: Properties of Types and Objects, Scalar Data Types, Composite Data Types **Encapsulation**: Structured Data Types, Abstract Data Types, Encapsulation by Subprograms, Type Definitions. **Inheritance**: Abstract Data Types Revisited, Inheritance, Polymorphism.

4. **Sequence Control**: Implement and Explicit Sequence Control, Sequence with Arithmetic Expressions, Sequence Control Between Statements, Sequencing with Non-arithmetic Expressions.

5. **Subprogram Control**: Subprogram Sequence Control, Attributes of Data Control, Parameter Transmission, Explicit Common Environment.

6. **Storage Management**: Elements Requiring Storage, Programmer- and System - Controlled Storage, Static Storage Management, Heap Storage Management.

7. **Distributed Processing**: Variations on Subprogram Control, Parallel Programming, Hardware Developments, Software Architecture. **Network Programming**: Desktop Publishing, The World Wide Web.

**Text Book:**

1. Programming languages – Design and Implementation by Terrence W. Pratt Marvin V.

Zelkowitz.3 rd Edition, Prentice Hall of India.

**Reference Books:**


1. Concepts of Programming Languages by Robert L. Sebesta, 4th Edition, Pearson Education.
2. Fundamentals of Programming Languages, Design & Implementation by Seyed H.Roosta. Vikas publications.
3. Programming Languages by Paradigm and Practice – Doris Appleby Julius J. Vendekopple Tata McGraw Hill Edition.

| CSE 3.1.6 | ELECTIVE-I | IMAGE PROCESSING |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course objectives**

1. To explain fundamentals of Image processing concepts.
2. To provide mathematical foundation of image enhancement , image compression and image segmentation.
3. To explain the students about Morphology and its applications in image processing.
4. To explain various methods and techniques for image transformation.

**Course outcomes**

1. Ability to develop algorithms for fundamental concepts in Image processing.
2. Ability to perform image enhancement , image compression and image segmentation using various methods.
3. Ability to implement Image transformation techniques

**Syllabus:**

1. **Fundamentals of Image Processing:** Image Acquisition, Image Model, Sampling, Quantization, Relationship Between Pixels, Distance Measures, Connectivity , Image Geometry, Photographic Film. Histogram: Definition, Decision Of Contrast Basing On Histogram, Operations Basing on Histograms Like Image Stretching, Image Sliding, Image Classification. Definition and Algorithm of Histogram Equalization.

2. **Image Enhancement in Spatial Domain** : Arithmetic and Logical Operations, Pixel or Point Operations, Size Operations; Smoothing Filters-Mean, Median, Mode Filters – Comparative Study.

3. **Edge enhancement in spatial domain:** Edge enhancement filters, Directorial Filters, Sobel, Laplacian, Robert, KIRSCH Homogeneity & DIFF filters, PREWITT Filter, Contrast based edge enhancement techniques, Comparative study, Low pass filters, High pass filters, Sharpening filters, Comparative study, Color fundamentals and color model

4. **Image Compression:** Run Length Encoding, modified run length encoding, Contour Coding, Huffman Code, Compression Due to Change in Domain, Compression Due to Quantization Compression at the Time of Image Transmission. Brief Discussion on:- Image Compression Standards.

5. **Image Segmentation:** Definition of segmentation, Characteristics of Segmentation, Detection of Discontinuities, Thresholding. Pixel Based Segmentation Method. Region Based Segmentation Methods, Segmentation by Pixel Aggregation, Segmentation by Sub Region Aggregation, Histogram Based Segmentation, Spilt and Merge Technique, Segmentation of moving objects.

6. **Morphology:** Dilation, Erosion, Opening, Closing, Hit-And-Miss Transform, Thinning, Thickening, Skeletons , Pruning Extensions to Gray – Scale Images Application of

Morphology in I.P

7. **Image Transforms :** A Detail Discussion On Fourier Transform, DFT,FFT, Properties of Fourier transform, WALSH Trans Form , WFT, HADAMARD Transform, DCT Image Enhancement in Frequency Domain: Design of Low Pass, High Pass, EDGE Enhancement, Smoothening Filters in Frequency Domain. Butter Worth Filter, Homomorphic Filters in Frequency Domain Advantages of Filters in Frequency Domain, Comparative Study of Filters in Frequency, Domain and Spatial Domain.

**Text Book:**

1. Digital Image Processing, Rafael C. Gonzalez And Richard E. Woods, Addision Wesley

**Reference Books:**

1. Fundamentals Of Electronic Image Processing By Arthyr – R – Weeks, Jr.(PHI)
2. Image Processing, Analysis, And Machine Vision By Milan Sonka Vaclan Halavac Roger Boyle, Vikas Publishing House.
3. Digital Image Processing, S. Jayaraman, S. Esakkirajan& T. Veera Kumar, TMH
4. Fundamentals of Digital Image Processing, Chris Solomon, Tobi Breckon, Wiley-Blackwell

| CSE 3.1.8 | DATABASE MANAGEMENT SYSTEMS LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** Univ. Exam: 3 Hours | | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**Course Objectives**:

1. To introduce to a commercial DBMS such as ORACLE.
2. To learn and practice SQL commands for schema creation, data manipulation.
3. To learn conceptual and physical database design based on a case study.
4. To apply database design stages by studying a case study.

**Course Outcomes**:

1. The student is exposed to a commercial RDBMS environment such as ORACLE.
2. The student will learn SQL commands for data definition and manipulation.
3. The student understands conceptual through physical data base design.
4. The student takes up a case study and applies the design steps.

Features of a commercial RDBMS package such as ORACLE/DB2, MS Access, MYSQL & Structured Query Language (SQL) used with the RDBMS.

I. Laboratory Exercises Should Include:

a. Defining Schemas for Applications,
b. Creation of Database,
c. Writing SQL Queries,
d. Retrieve Information from Database,
e. Creating Views
f. Creating Triggers
g. Normalization up to Third Normal Form
h. Use of Host Languages,
i. Interface with Embedded SQL,
j. Use of Forms
k. Report Writing

II. Some sample applications are given below:

1. Accounting Package for Shops,
2. Database Manager for Magazine Agency or Newspaper Agency,
3. Ticket Booking for Performances,
4. Preparing Greeting Cards & Birthday Cards
5. Personal Accounts - Insurance, Loans, Mortgage Payments, Etc.,
6. Doctor's Diary & Billing System
7. Personal Bank Account
8. Class Marks Management
9. Hostel Accounting
10. Video Tape Library,
11. History of Cricket Scores,
12. Cable TV Transmission Program Manager,
13. Personal Library.
14. Sailors Database
15. Suppliers and Parts Database

| CSE 3.1.9 | DATA COMMUNICATIONS AND COMPUTER NETWORKS LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** Univ. Exam: 3 Hours | | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**FIRST CYCLE OF EXPERIMENTS**

1.1 PC-to-PC COMMUNICATIONS UNDER DOS WITH NULL MODEM a) Using Serial Ports and RS-232 C Cable Connection b) Using Paralell Ports and Parallel Cable Connection

1.2 PC-to-PC COMMUNICATIONS UNDER DOS WITH MODEM and 4-LINE EXCHANGE Using Communication Software: COMIT or XTALK

1.3 PC-to-PC COMMUNICATIONS UNDER WIN 98's DIRECT CABLE CONNECTION with NULL MODEM a) Using Serial Ports and RS-232 C Cable Connection b) Using Paralell Ports and Parallel Cable Connection

1.4 PC-to-PC COMMUNICATIONS UNDER WIN 98's DIAL-UP NETWORKING WITH MODEM and 4-LINE EXCHANGE

1.5 PC-to-PC COMMUNICATIONS UNDER WIN 98's HYPER TERMINAL WITH MODEM and 4-LINE EXCHANGE

1.6 a) LAN WITH BUS TOPOLOGY with a minimum of two systems i) Windows Peer-to-Peer Network ii) Windows NT Client-Server Network b) LAN WITH STAR TOPOLOGY with a minimum of two systems

1.7 a) LAN WITH BUS TOPOLOGY with a minimum of two systems using NOVELL Netware b) LAN WITH STAR TOPOLOGY with a minimum of two systems using NOVELL Netware

**SECOND CYCLE OF EXPERIMENTS**

2.1 INERNET CONNECTION SET-UP USING DIAL-UP NETWORKING

2.2 TERMINAL NETWORK WITH UNIX/LINUX SERVER and one or two Terminals

2.3 TERMINAL NETWORK WITH UNIX/LINUX SERVER, Terminal Server, and one or two terminals

2.4 NETWORK PROGRAMMING EXERCISE-I USING A SIMPLIFIED API Echo software( Develop echo client and echo server programs and run the two programs on separate computers and verify that they can communicate Chat software (Develop chat client and chat server programs and test to ensure they can communicate). Build a simple file transfer service that consists of client and server

2.5 NETWORK PROGRAMMING EXERCISE -II USING THE SOCKET API Write an echo client and server using sockets Build a web server using sockets

2.6 CONCURRENT NETWORK PROGRAMMING EXERCISE –III Build a Concurrent server(threads) – Create a server capable of handling connections from multiple clients concurrently Build a Concurrent file transfer server(processes) – Create separate processes to allow a server to handle multiple clients concurrently

2.7 NETWORK PROGRAMMING EXERCISE –IV USING PROTOCOL DESIGN Design a reliable data transfer protocol ( Devise, implement and test a protocol that provides reliable data transfer across a network that drops, delays or corrupts packets Design stop and wait flow control protocol Design a sliding window protocol 2.7.1 NETWORK PROGRAMMING EXERCISE –V WITH PROTOCOLS FROM TCP/IP SUITE Build a domain name system client program 69

| CSE 3.1.10 | SOFT SKILLS LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** | Univ. Exam: 3 Hours | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

1. English Language Skills
2. Spoken English Skills
3. Presentation Skills

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING(A) - VISAKHAPATNAM**
**II – SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION**
**Branch: COMPUTER SCIENCE AND ENGINEERING**
**III/IV B.TECH** (FOUR YEAR COURSE) &
**III/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)
(With effect from **2015-2016** admitted batch onwards)
Under Choice based Credit System

| **B.TECH. (CSE) 3rd YEAR II-SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION WITH EFFECT FROM 2015-2016 20115ADMITTED BATCHES** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **SUB.REF** | **NAME OFTHE SUBJECT** | **PERIODS** | | | **MAXIMUM MARKS** | | | **CREDITS** |
| | | **THEORY** | **TUTORIAL** | **LAB** | **EXAM** | **SESSIONALS** | **TOTAL** | |
| CSE 3.2.1 | DATA WAREHOUSING AND DATA MINING | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.2.2 | OBJECT ORIENTED SOFTWARE ENGINEERING | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.2.3 | MOOCS-III | -- | -- | -- | -- | -- | 100 | 2 |
| CSE 3.2.4 | DESIGN AND ANALYSIS OF ALGORITHMS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.2.5 | ELECTIVE–II | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.2.6 | COMPILER DESIGN | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.2.7 | CRYPTOGRAPHY AND NETWORK SECURITY | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 3.2.8 | SOFTWARE ENGINEERING MINI PROJECT LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 3.2.9 | WEB TECHNOLOGIES LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| **TOTAL CREDITS** | | | | | | | | 30 |

**ELECTIVE-II**

1) Cloud Computing 2) Soft Computing 3) Distributed Systems 4) Advanced Computer Architecture 5) Computer Graphics

**List of topics for MOOCS-III:**

1)Internet of Things 2) Rasberry Pi Platform 3) Signals and Systems 4) DevOps 5) Social Network Analysis/Mining

| CSE 3.2.1 | DATAWAREHOUSING & DATA MINING | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives** :

1. To understand the evolution of data warehousing and data mining systems
2. To understand extracting, cleaning and transformation of data into a warehouse.
3. To learn the principles of statistics, information theory, machine learning and other areas AI and implementation of data mining techniques.
4. To understand pattern mining using classification and clustering methods.

**Course Outcomes:**

1. The student understands the differences between OLTP and OLAP.
2. The student learns how data cube technology supports summarization and querying high dimensional data.
3. The student is introduced to similarity, distance, information gain and other performance and error metrics used for evaluation of mining results.
4. The student is introduced to various approaches to association rule mining , supervised and unsupervised learning and the corresponding classification and clustering approaches involving decision trees, Bayesian approaches, model based and agglomerative approaches.

**Syllabus:**

1. **Introduction to Data Mining:** Evolution of IT into DBMS, Motivation and importance of Data Warehousing and Data Mining, Kinds of Patterns, Technologies,Applications, Major Issues in Data Mining, Data Objects and Attributes Types, Statistical Descriptions of Data, Data Visualization, Estimating Data Similarity and Dissimilarity,
2. **Data pre-processing :** Quality data, Data Cleaning, Data Integration, Data Reduction, Data Transformation, Discretization and Concept Hierarchy Generation.
3. **Data Warehouse and OLAP Technology:** Basic Concepts of Data warehouse, Data Modeling using Cubes and OLAP, DWH Design and usage, Implementation using Data Cubes and OLAPs, Data Generalization with AOI.
4. **Data Cube Technology:** Preliminary Concepts of Data Cube Computation, Data Cube Computation Methods: Multi-way Array Aggregation for Full Cube, Multi-dimensional Data Analysis in cube space,
5. **Mining Frequent Patterns Based on Associations and Correlations**: Basic Concepts, Frequent Itemset Mining Methods: Apriori Algorithm, Association Rule Generation, Improvements to A Priori, FP- Growth Approach, Mining Closed and Max Patterns, Pattern Evaluation Methods, Association mining in multi-level, multi-dimensional space

6. **Classification & Prediction**: Basic Concepts, Decision Tree Induction, Bayes Classification, Rule- Based Classification, Model Evaluation and Selection, Techniques to Improve Classification Accuracy, Classification by Back Propagation, Associative Classification, K-nearest neighbor classifier.
7. **Cluster Analysis:** Basic Concepts and issues in clustering, Types of Data in Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Evaluation of Clustering Solutions

**Text Book:**

1. Data Mining- Concepts and Techniques by Jiawei Han, Micheline Kamber and Jian Pei –Morgan Kaufmann publishers –--3rd edition

**References:**

1. Introduction to Data Mining, Adriaan, Addison Wesley Publication
2. Data Mining Techniques, A.K.Pujari, University Press Data mining concepts by Tan, Steinbech, and Vipin Kumar - Pearson Edu publishers

| CSE 3.2.2 | OBJECT ORIENTED SOFTWARE ENGINEERING | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course objectives:**

1. To explain the importance of OOSE in Software development.
2. To explain the students the importance of Requirements Engineering.
3. To explain the role of UML and Testing in Software Development.
4. To explain the entire Software Development Process with aid of case studies.

**Course Outcomes:**

1. Ability to define a problem and perform Requirements Engineering.
2. Ability to draw UML diagrams for the requirements gathered.
3. Ability to implement the designed problem in Object Oriented Programming Language and
4. test whether all the requirements specified have been achieved or not.

1. **Introduction to Object Oriented Software Engineering:** Nature of the Software, Types of Software , Software Engineering Projects, Software Engineering Activities, Software Quality, Introduction to Object Orientation, Software Process Models- Waterfall Model, Opportunistic Model , Phased Released Model, Spiral Model, Evolutionary Model, Concurrent Engineering Model

2. **Requirements Engineering:** Domain Analysis, Problem Definition and Scope, Requirements Definition, Types of Requirements, Techniques for Gathering and Analyzing Requirements, Requirement Documents, Reviewing, Managing Change in Requirements.

3. **Unified Modeling Language & Use Case Modeling:** Introduction to UML, Modeling Concepts, Types of UML Diagrams with Examples; User-Centered Design, Characteristics of Users, Developing Use - Case Models of Systems, Use-Case Diagram, Use- Case Descriptions, Basics of User Interface Design, Usability Principles, User Interfaces.

4. **Class Design and Class Diagrams:** Essentials of UML Class Diagrams, Associations and Multiplicity, Other Relationships, Generalization, Instance Diagrams, Advanced Features of Class Diagrams, Interaction and Behavioral Diagrams: Interaction Diagrams, State Diagrams, Activity Diagrams, Component and Deployment Diagrams.

5. **Software Design and Architecture:** Process of Design, Principles Leading to Good

Design, Techniques for Making Good Design Decisions, Good Design Document; Pattern Introduction, Design Patterns: Abstraction-Occurrence Pattern, General Hierarchical Pattern, Play-Role Pattern, Singleton Pattern, Observer Pattern, Delegation Pattern, Adaptor Pattern, Façade Pattern, Immutable Pattern, Read-Only Interface Pattern and The Proxy Pattern; Software Architecture Contents of Architecture Model, Architectural Patterns: Multilayer, Client-Server, Broker, Transaction Processing, Pipe & Filter and MVC Architectural Patterns

6. **Software Testing:** Overview of Testing, Testing Concepts, Testing Activities, Testing Strategies, Unit Testing, Integration Testing, Function Testing, Structural Testing, Class Based Testing Strategies, Use Case/Scenario Based Testing, Regression Testing, Performance Testing, System Testing, Acceptance Testing, Installation Testing, OO Test Design Issues, Test Case Design, Quality Assurance, Root Cause Analysis, Post-Mortem Analysis.

7. **Software Process Management:** Introduction to Software Project Management, Rationale Management, Configuration Management, Activities of Software Project Management, Structure of Project Plan, Software Engineering Teams, Software Cost Estimation, Project Scheduling, Tracking and Monitoring.

**CASE STUDY:**

1. Simple Chat Instant Messaging System
2. GPS Based Automobile Navigation System
3. Waste Management Inspection Tracking System (WMITS)
4. Geographical Information System

**Text Book:**


1. Object-Oriented Software Engineering Practical software development using UML and Java by Timothy C. Lethbridge & Robert, Langaniere Mcgraw-Hill
2. Object-Oriented Software Engineering: Using UML, Patterns and Java, Bernd Bruegge and Allen H. Dutoit, 2nd Edition, Pearson Education Asia.


**Reference:**


1. Software Engineering: A Practitioner's Approach, Roger S Pressman.
2. A Practical Guide to Testing Object-Oriented Software, John D. McGregor; David A. Sykes, Addison-Wesley Professional.
3. Software Engineeing, K.K. Agarwal, New Age Publications 2008

| CSE 3.2.4 | DESIGN & ANALYSIS OF ALGORITHMS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

On completing this course student will be able to

1. Solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound and writing programs for these solutions
2. Analyze the asymptotic performance of algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Synthesize efficient algorithms in common engineering design situations.

**Course Outcomes :**

1. Students will be able to justify the correctness of algorithms using inductive proofs and invariants
2. Analyze worst-case running times of algorithms using asymptotic analysis.
3. Describe various paradigms of design use them appropriately when an algorithmic design situation calls for it.
4. Students will be able to Compare between different data structures. Pick an appropriate data structure for a design situation.

**Syllabus:**

1. **Introduction –** Fundamentals of algorithmic problem solving – important problem type. Fundamentals of analysis of algorithms and efficiency – Analysis framework – Asymptotic Notations and Basic Efficiency classes – Mathematical Analysis of Non- recursive Algorithms – Mathematical Analysis of recursive Algorithms – Empirical Analysis of Algorithms – Algorithm Visualization

2. **Brute Force** – Selection Sort and Bubble sort – Sequential Search and Brute – Force String Matching – Closest Pair and Convex-Hull Problems by Brute Force – Exhaustive Search **Divide-and-Conquer** – Merge sort – Quick sort – Binary Search – Binary Tree Traversals and Related Properties – Multiplication of large integers and Strassen's Matrix Multiplication – Closest- Pair Convex-Hull Problems by Divide- and – Conquer

3. **Decrease – and – Conquer** – Insertion Sort – Depth-First Search and Breadth-First Search- Topological Sorting – Algorithms for Generating Combinatorial Objects – Decrease-by-a- Constant-Factor Algorithms – Variable-Size-Decrease Algorithms.

4. **Transform-and-Conquer** – Presorting – Gaussian Elimination – Balanced Search Trees – Heaps and Heap sort – Horner's Rule and Binary Exponentiation – Problem Reduction

   **Space and Time Tradeoffs** – Sorting by Counting – Input Enhancement in string Matching – Hashing – B-Trees

5. **Dynamic Programming** – Computing a Binomial Coefficient – Warshall's and Floyd's Algorithm – Optimal Binary Search Trees - The Knapsack Problem and Memory

Functions

6. **Greedy Technique** – Prim's Algorithm – Kruskal's Algorithm – Dijkstra's Algorithm – Huffman Trees **Limitations of Algorithm Power** – Lower-Bound Arguments – Decision Trees – P, NP and NP – complete problems – Challenges of Numerical Algorithms
7. **Coping with the Limitations of Algorithms Power** – Backtracking – Branch-and-Bound – Approximation Algorithms for NP-hard Problems – Algorithms for solving Nonlinear Equations.

**Text Book:**


1. Introduction to Design & Analysis of Algorithms by Anany Levitin, Pearson Education, New Delhi, 2003
2. Fundamentals of Computer Algorithms, Horowitz and Sahni, Galgothia publications.


**Reference Books:**


1. Introduction to Algorithms by Thomas H. Corman, Charles E. Leiserson, Ronald R. Rivest & Clifford Stein, Prentice Hall of India, New Delhi, New Delhi.

| CSE 3.2.5 | ELECTIVE II | CLOUD COMPUTING |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** | Univ. Exam: 3 Hours | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To import fundamental concepts in the area of cloud computing.
2. To understand the concept of Virtualization and cloud data storage.
3. To learn cloud Application Development and cloud Governance.
4. To gain competence in Map Reduce and Hadoop Overview.

**Course Outcomes:**

1. Identify the architecture and infrastructure of cloud computing.
2. Develop applications for cloud computing.
3. Design and Implement a novel cloud computing application.

**Syllabus**

1. **Introduction to cloud computing:** Cloud computing components, Infrastructure services, storage applications, database services – introduction to Saas, Paas, Iaas, Idaas, data storage in cloud

2. **Virtualization:** enabling technologies, types of virtualization, server virtualization, desktop virtualization, memory virtualization, application and storage virtualization-tools and products available for virtualization

3. **SAAS and PAAS:** Getting started with Saas, SaaS solutions,SOA , PaaS and benefits.

4. **Iaas and Cloud data storage:** understanding Iaas, improving performance for load balancing, server types within Iaas, utilizing cloud based NAS devices, cloud based data storage, and backup services, cloud based block storage and database services

5. **Cloud Application development:** Client server distributed architecture for cloud designing cloud based solutions, coding cloud based applications,traditional Apps vs cloud Apps, client side programming, server side programming overview-fundamental treatment of web application frameworks.
6. **Cloud Governance and economics:** Securing the cloud, disaster recovery and business continuity in the cloud, Managing the cloud, migrating to the cloud, governing and evaluating the clouds business impact and economics,

7. **Inside Cloud**: Introduction to MapReduce and Hadoop-over view of big data and its impact on cloud

**Text Books:**

1. Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile,

Security and More, Kris Jamsa, Jones & Bartlett Publishers,Paper back edition, 2013
2. Cloud Computing: A Practical Approach,Anthony T .Velte, Toby J.Velte, Robert Elsenpeter, Tata McGraw Hill Edition

**References:**

1. Hadoop Map Reduce cookbook, SrinathPerera and Thilina Gunarathne, Packt publishing

| CSE 3.2.5 | ELECTIVE II | SOFT COMPUTING |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course objectives:**

1. To make the student to understand the role of imprecision and uncertainty in real world scenarios.
2. To explain the role of Soft Computing in addressing the imprecision and uncertainty.
3. To explain the principal components of soft computing that include Fuzzy Sets and Fuzzy Logic, Artificial Neural Networks, Genetic Algorithms and Rough Sets.
4. To learn the Design and Implementation of Soft Computing methodologies.
5. To explain the design of hybrid systems which is combination of one or more soft computing methodologies mentioned.

**Course outcomes:**

1. Ability to represent Uncertainty / imprecision data.
2. Ability to select a suitable method of Soft Computing to solve a particular problem.
3. Ability to build hybrid systems using Soft Computing techniques.

**Syllabus:**

1. **Soft Computing:** Introduction to Fuzzy Computing, Neural Computing, Genetic Algorithms, Associative Memory, Adaptive Resonance Theory, Different Tools and Techniques, Usefulness and Applications.

2. **Fuzzy Sets and Fuzzy Logic**: Introduction, Fuzzy Sets Versus Crisp Sets, Operations on Fuzzy Sets, Extension Principle, Fuzzy Relations and Relation Equations, Fuzzy Numbers, Linguistic Variables, Fuzzy Logic, Linguistic Hedges, Applications,

3. **Interference in fuzzy logic:** fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzifications and Defuzzificataions, Fuzzy Controller, Fuzzy Controllers, Fuzzy Pattern Recognition, Fuzzy Image Processing, Fuzzy Database.

4. **Artificial Neural Network**: Introduction, Artificial Neuron and its model, activation functions, Neural network architecture: single layer and multilayer feed forward networks, re-current networks. Various learning techniques, perception and convergence rule, Auto-associative and hetro-associative memory , Hebb's Learning, Adaline, Perceptron

5. **Multilayer Feed Forward Network:** Back Propagation Algorithms, Different Issues Regarding Convergence of Multilayer Perceptron, Competitive Learning, Self-Organizing, Feature Maps, Adaptive Resonance Theory, Associative Memories, Applications.

6. **Evolutionary and Stochastic Techniques:** Genetic Algorithm (GA), Genetic Representations, (Encoding) Initialization and Selection, Different Operators of GA, Analysis of Selection Operations, Hypothesis of Building Blocks, Schema Theorem and Convergence of Genetic Algorithm, Simulated Annealing and Stochastic Models, Boltzmann Machine, Applications.

7. **Rough Set:** Introduction, Imprecise Categories Approximations and Rough Sets, Reduction of Knowledge, Decision Tables and Applications. Hybrid Systems: Neural-Network-Based Fuzzy Systems, Fuzzy Logic-Based Neural Networks, Genetic Algorithm for Neural Network Design and Learning, Fuzzy Logic and Genetic Algorithm for Optimization, Applications

**Text Books:**


1. Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications, S. Rajsekaran and G.A. Vijayalakshmi Pai, Prentice Hall of India.
2. Rough Sets, Z.Pawlak, Kluwer Academic Publisher, 1991.
3. Intelligent Hybrid Systems, D. Ruan, Kluwer Academic Publisher, 1997


**References:**


1. Artificial Intelligence and Intelligent Systems, N.P.Padhy, Oxford University Press.
2. Neural Fuzzy Systems, Chin-Teng Lin & C. S. George Lee, Prentice Hall PTR. Addison-Wesley
3. Learning and Soft Computing, V. Kecman, MIT Press, 2001
4. Fuzzy Sets and Fuzzy Logic, Klir & Yuan, PHI, 1997

| CSE 3.2.5 | ELECTIVE II | DISTRIBUTED SYSTEMS |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

This course provides an introduction to the fundamentals of distributed computer systems, assuming the availability of facilities for data transmission.

**Course Outcomes:**

By the end of the course, students should be able to build distributed systems that:

1. Scale as the number of entities in the system increase
2. Can sustain failures and recover from them
3. Work with distributed, fault tolerant file systems
4. Can handle and process large data volumes
5. Are secure and handle certain classes of distributed denial of service attacks
6. Are Loosely coupled, transactional and eventually stable

**Syllabus:**

1. Introduction to Distributed Systems, What is a Distributed System?, Hard ware concepts, Software concepts, Design issues.
2. Communication in Distributed Systems, Lay red Protocols, ATM networks, The Client – server model, Remote Procedure call, Group communication.
3. Synchronization in Distributed System, Clock Synchronization, Mutual Exclusion, Election algorithms, Atomic transactions, Deadlocks in Distributed Systems.
4. Process and processors in Distributed System threads, System Models, Processors allocation, Scheduling in Distributed System, Fault tolerance, Real time Distributed System.
5. Distributed File Systems, Distributed File System Design, Distributed File System implementation, Trends in Distributed File System.
6. Distributed Shared Memory, Introduction, What is Shared memory?, Consistency models, Page based Distributed Shared memory, Shared – variable Distributed Shared memory, Object based Distributed Shared Memory.

**Text Book:**

Distributed Operating Systems, Andrew S. Tanenbanm

**Reference Book:**

Advanced Concepts in Operating Systems, Makes Singhal and Niranjan G.Shivaratna

| CSE 3.2.5 | ELECTIVE II | ADVANCED COMPUTER ARCHITECTURE |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. Study about multiprocessors, shares memory, distributed computers, vector super computers.
2. Learn about advanced processing , pipelining and parallel and multi threaded architectures.
3. Study of parallel programming models, languages, code optimization pipelining.
4. Study about multiprocessor , multithreaded and multicomputer unix design.

**Course Outcomes:**

1. Detailed idea about parallel computing models.
2. Knowledge on hand about advanced processing, pipelining and super scalar techniques , Parallel, multi vector and multithreaded architectures.
3. Knowing about parallel programming softwares.

**Syllabus:**

1. **Parallel Computer Models:** Elements of Modern Computer, Multi Processor and Multi Computers- Shared Memory Multi processors, Distributed Memory Multi Computers, Multi Vector and SIMD Computers-Vector Super Computers and SMD Super Computers, Architectural Development Tracks- Multiple Processor Tracks, Multi Vector and SIMD Tracks

2. **Advanced Processing:** Advance Processing Technology-Design Space of Processor, Instruction Set Architectures, CISC Scalar Processors, RISC Scalar Processors; Super Scalar and Vector Processor- Super Scalar Processors, VLIW Architecture, Vector and Symbolic Processors.

3. **Pipelining and Super scalar Techniques:** Linear Pipeline Processors, Non-linear pipelining Processors, Instruction Pipeline Design, Arithmetic Pipeline Design, Superscalar and Super pipeline Design.

4. **Parallel Architectures**: Multiprocessor Interconnects- Hierarchical Bus Systems, Crossbar Switch and Multiport Memory, Multistage and Combining Networks; Cache Coherence and Synchronization Mechanisms,- The Cache Coherence Problem, Snoopy bus Protocols, Directory based protocols, Hardware Synchronization Mechanisms.

5. **Multivector and SIMD Computer:** Vector Processing Principles- Vector Instruction Types, vector- Access Memory Schemes, past and Present Supercomputers; Multi Vector Multiprocessors- Performance-Directed Design Rules, Cray Y-MP, C-90, and MPP, Mainframes and Mini- supercomputers; SIMD Computer Organizations- Implementation Models, the CM-2 Architecture, The Maspar MP-1 Architecture.

6. **Multithreaded Architecture:** Latency Hiding techniques- Shared Virtual Memory, Perfecting Techniques, Distributed Coherent Caches, Scalable Coherence Interface, relaxed Memory Consistency. Principles of Multithreading- multithreading Issues and Solutions, Multiple Context Processors, Multidimensional Architectures.

7. **Software for Parallel Programming:** Parallel programming Models, parallel languages and Compilers, Code optimization and Scheduling, Loop parallelization and Pipelining.

8. **Case Study:** Multiprocessor UNIX Design Goals, Master-Slave and Multithreaded UNIX, Multicomputer UNIX Extensions.

**Textbook:**

1. Kai Hwang, Advanced Computer Architecture, Tata McGraw Hill Edition.

**References:**

1. Kai Hwang and Briggs, Advanced Computer Architecture and Parallel Processing, Tata McGraw Hill Edition.
2. Advanced Computer Architectures: A Design Space Approach Sima, Terence FOUNTAIN, Péter KACSUK Pearson Education India.
3. *Computer Architecture: A Quantitative Approach, 5th Edition.* John L. Hennessy,David A. Patterson, Morgan Kaufmann Publications.

| CSE 3.2.5 | ELECTIVE II | COMPUTER GRAPHICS | |
|---|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** | | Univ. Exam: 3 Hours | **Credits: 4** |
| **Internal: 30 Marks** | | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. Provides a comprehensive introduction to computer graphics with a foundation in Graphics Applications.
2. A thorough introduction to computer graphics techniques.
3. To give the basics of Geometric Transformations and projections.
4. To introduce three dimensional concepts and object representations with color models and basics of computer animation.

**Course Outcomes:**

1. The students will understand graphics principles and graphics hardware.
2. The students can demonstrate geometrical transformations.
3. The students can create interactive graphics applications and demonstrate computer graphics animation.

**Syllabus:**

1. **Introduction:** Computer Graphics and their applications: Computer Aided Design, Computer Art, Entertainment, Education and Training, Graphical User Interfaces; Overview of Graphics systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Graphics Monitors And Workstations, Input Devices, Hard Copy Devices, Interactive Input Methods, Windows and Icons, Virtual Reality Environments,Graphics Software.

2. **Output primitives :**Points and Lines, **,** Line and Curve Attributes, Color and Gray scale levels, Antialiasing, Loading the Frame buffer, Line function, Line Drawing Algorithms, Circle Generating Algorithms, Ellipse Generating Algorithms, Pixel Addressing, Area Fill Attributes, Filled Area Primitives, Filled Area Functions, Cell Array, Character Generation, Character Attributes, Bundled Attributes, Curve Functions, Parallel Curve Algorithms.

3. **Two Dimensional Transformations:** Basic 2D Transformations, Matrix Representations, Homogeneous Coordinates, Composite Transformations, Other Transformations, Transformations between Coordinate Systems, Affine Transformations.

4. **Three Dimensional Transformations & Projections:** Translation, Rotation, Scaling, Other Transformations, Composite Transformations, 3D Transformation Functions, Modeling and Coordinate Transformations, Need for projections, Parallel & Perspective projections, General Projection Transformations.

5. **Viewing Pipeline and Clipping operations :** Viewing Pipeline ,Viewing Coordinates & Reference frames, Window-to-Viewport Coordinate Transformation, Two Dimensional Viewing Functions, , Three Dimensional Viewing, View Volumes, Clipping and its Operations, Types of clipping operations- Point Clipping,Line Clipping, Polygon Clipping,, Curve Clipping,, Text and Exterior Clipping.

6. **Three Dimensional Concepts and Object representations:** 3D display methods, 3D Graphics, Polygon Surfaces, Curved Lines and Surfaces, Quadratic Surfaces, Super Quadrics, Blobby Objects, Spline Representations, Cubic Spline methods, Bézier Curves and Surfaces, B-Spline Curves and Surfaces,

7. **Color Models and Basics of Computer Animation:** Intuitive color concepts, Basics of RGB Color model, YIQ Color Model, CMY & HSV Color models. Design of animation Sequences, Raster Animations, Key Frame systems: Morphing, A Simple program on Animation.

**Text Book:**

1. Computer Graphics, Donald Hearn & M.Pauline Baker, Pearson Education ,New Delhi.

**Reference Books:**

1. ProceduralElementsforComputerGraphics,DavidF.Rogers,TataMcGrawHillBookCompany,NewDelhi, 2003
2. ComputerGraphics:Principles&PracticeinC,J.D.Foley,S.KFeiner,AVanDamF.HJohnPearsonEducation,20 04
3. ComputerGraphicsusingOpenGL,FranscisSHillJr,PearsonEducation,2004.
4. Computer Vision and Image Processing: A Practical Approach using CVIPtools, S. E. Umbaugh,, Prentice Hall, 1998

| CSE 3.2.6 | COMPILER DESIGN | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course objectives:**

1. To explain the basic understanding of grammars and language definition and Introducing various phases of designing a compiler.
2. To make the student to understand the concepts underlying the design and implementation of language processors and its mechanisms.
3. To extend the knowledge of parser by parsing LL parser and LR parser.
4. To enrich the knowledge in various phases of compiler ant its use, code optimization techniques, loop optimization techniques, machine code generation, and use of symbol table..

**Course outcomes:**

1. Ability to design & conduct experiments for Intermediate Code Generation in compiler.
2. Ability to learn the new code optimization techniques to improve the performance of a program in terms of speed & space.
3. Ability to acquire the knowledge of modern compiler & its features.

**Syllabus**

1. **Introduction:** Introduction to Compilers and Language processors, , Programming Language basics, Structure & Different Phases of a Compiler, Review of Compiler Structure, Structure of Optimizing Compilation, Compiler construction tools, Boot strapping, Cross compilers.
2. **Finite Automata & Lexical Analysis:** Introduction to Lexical Analysis, Lexical Analyzers, Approaches to design Lexical Analyzers, Language for specifying lexical analyzers, Introduction to Finite automata, Regular Expressions & Languages, Recognition of Tokens, Transition Diagrams, Implementation of lexical analyzers, Lexical Analyzer Generator LEX.
3. **Syntax Analysis:** Syntactic Specification of Programming Languages, Context Free Grammars & Languages, Introduction to Parsers. Top-down parsing techniques: Brute force parsing, Recursive Descent Parsing, Predictive Parsing, Bottom – up Parsing: Shift reduce parsing,Operator parsing, LR (k) parsing.
4. **Semantic Analysis and Intermediate Code Generation:** Semantic Actions, Syntax Directed Translations, Translation on the parse Tree, Implementation of Syntax Directed Translator, Intermediate Codes, Syntax Directed translation to Postfix code, Syntax Trees, Intermediate Code Generation, Three Addr5ess Code-Translation of Expressions, TypeChecking& Type Conversions.
5. **Code Optimization:** Principal sources of Code Optimization, Loop Optimization, Basic Blocks& Flow Graphs, DAG Representation of Basic Blocks, Applications of DAG, Local Optimization, ,Unreachable Code Elimination, Dead Code El imination ,Data Flow Analysis, Data Flow Equations & Computations, Peep-Hole

Optimization. Machine Dependent Optimizations, Overview of Informal Compiler Algorithm Notation(ICAN), If Simplification, Loop Simplification, Loop Inversion, Branch Optimization and Prediction

6. **Code Generation and Code Scheduling:** Issues in Code Generation, Input to Code Generator, Instruction Selection, Register Allocation, Simple Target Machine Model, Program and Instruction Costs, Register allocation & Assignments, Code Generation Algorithm, Code Generators, Optimal Code Generation for Expressions, Code Generation From DAG.

7. **Symbol Tables, Run time Environment and Error Handling**: Contents of a Symbol Table, Data Structures for Symbol Tables; Run time Environments, Implementation of a simple Stack allocation, Heap Management, Block Structured Languages; Error Detection & Recovery, Lexical Phase Errors, Syntactic & Semantic Errors, Error Handling Routines.

**Text Books:**


1. Principles of Compiler Design by Aho,D. Ullman, Lam and Ravi Sethi, Pearson Education Second Edition
2. Advanced Compiler Design and Implementation, Steven Muchnic, Elsevier Publications.


**Reference Books**:


1. Compiler Construction by Kenneth. C. Louden, VikasPub.House.
2. Compiler Design, A.A. Pentambekar, TechnicalPublications
3. Modern Compiler Design, Grune.D, Van Reeuwijk K, Bal H.E, Jacobs C J H, Langendoen K, Springer,

| CSE 3.2.7 | CRYPTOGRAPHY & NETWORK SECURITY | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. Introduction of the issues in network security- its need and importance, taxonomy and terminology.
2. Discussion of various cryptographic techniques.
3. Exploration of different types of security threats and remedies.
4. Understanding of Internet security protocols and standards

**Course Outcomes:**

1. Realize the need and importance of network and data security in the Internet and in the distributed environments.
2. Identify the different types of network security issues and their remedies.
3. Application various cryptographic tools and techniques in different contexts and as per need of security levels.
4. Implementation of some Internet security protocols and standards

**Syllabus:**

1 **Overview:** Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, A Security Architecture for Open Systems, Computer Security Trends, Computer Security Strategy. Cryptographic Tools: Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data. User Authentication: Means of Authentication, Password-Based Authentication, Token-Based Authentication, Biometric Authentication, Remote User Authentication, Security Issues for User Authentication, Practical Application: An Iris Biometric System, Case Study: Security Problems for ATM Systems.

2 **Access Control:** Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Case Study: RBAC System for a Bank. Database Security: The Need for Database Security, Database Management Systems, Relational Databases, Database Access Control, Inference, Statistical Databases, Database Encryption, Cloud Security.

3 **Malicious Software:** Types of Malicious Software (Malware), Propagation—Infected Content—Viruses, Propagation—Vulnerability Exploit—Worms, Propagation—Social Engineering—SPAM E-mail, Trojans, Payload—System Corruption, Payload—Attack Agent—Zombie, Bots, Payload—Information Theft—Key loggers, Phishing, Spyware, Payload—Stealthing—Backdoors, Root kits, Countermeasures.
Denial-of-Service Attacks: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Application-Based Bandwidth Attacks, Reflector and Amplifier Attacks, Defenses Against Denial-of-Service Attacks, Responding to a Denial-of-Service Attack.

4 **Intrusion Detection:** Intruders, Intrusion Detection, Host-Based Intrusion Detection, Distributed Host-Based Intrusion Detection, Network-Based Intrusion Detection, Distributed Adaptive Intrusion Detection, Intrusion Detection Exchange Format, Honeypots, Example System: Snort. Firewalls and Intrusion Prevention Systems: The Need for Firewalls, Firewall Characteristics, Types of Firewalls, Firewall Basing, Firewall Location and Configurations, Intrusion Prevention Systems, Example: Unified Threat Management Products.

5 **Buffer Overflow:** Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks, Software Security: Software Security Issues, Handling Program Input, Writing Safe Program Code, Interacting with the Operating System and Other Programs,

Handling Program Output. Operating System Security: Introduction to Operating System Security, System Security Planning, Operating Systems Hardening, Application Security, Security Maintenance, Linux/Unix Security, Windows Security, Virtualization Security.

6 **Symmetric Encryption and Message Confidentiality:** Symmetric Encryption Principles, Data Encryption Standard, Advanced Encryption Standard, Stream Ciphers and RC4, Cipher Block Modes of Operation, Location of Symmetric Encryption Devices, Key Distribution. Public-Key Cryptography and Message Authentication: Secure Hash Function, HMAC, The RSA Public-Key Encryption Algorithm, Diffie-Hellman and Other Asymmetric Algorithms.

7 **Internet Security Protocols and Standards:** Secure E-mail and S/MIME, Domain Keys Identified Mail, Secure Socket Layer (SSL) and Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security. Internet Authentication Applications: Kerberos, X.509, Public-Key Infrastructure, Federated Identity Management. Wireless Network Security: Wireless Security Overview, IEEE 802.11 Wireless LAN Overview, IEEE 802.11i Wireless LAN Security.

**Text Book:**

1. Computer Security - Principles and Practices (Except the Chapters 13, 14, 15, 16, 17, 18, 19), 2nd Edition by William Stallings, Pearson Education, Inc.

**Reference Books:**

1. Cryptography and Network Security by William Stallings, Pearson Education Asia, New Delhi.
2. Network Security Essentials Applications and Standards, by William Stallings, Pearson Education Asia, New Delhi.

| CSE 3.2.8 | SOFTWARE ENGINEERING MINI PROJECT LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** Univ. Exam: 3 Hours | | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

The purpose of the Software Engineering Lab course is to familiarize the students with modern software engineering methods and tools, **Rational Products**. The course is realized as a project-like assignment that can, in principle, by a team of three/four students working full time. Typically the assignments have been completed during the semester by each project team.

The goal of the Software Engineering Project is to have a walk through from the requirements, design to implementing and testing. An emphasis is put on proper documentation. Term projects are projects that a group student might take through from initial specification to implementation by giving equal importance to both design and implementation.

**Cycle I: Practicing UML diagrams using IBM Rational Rose.** 6*3 periods= 18periods

Before developing a mini-project, in this cycle, the student is acquainted with different UML diagrams using Rational Rose. The experiments should include drawing UML diagrams listed below for two demo/example applications assigned by the lab Instructor. The input for the following experiments is problem statement for any two demo projects supplied by the instructor.

1. Introduction to Rational Rose and Practicing the following diagrams
    a. Activity diagrams for the overall business process of the projects
    b. Use-case diagram for the demo projects along with Use-case descriptions and sub-diagrams for Use-cases.

2. Class diagram- Class diagrams including the features like classes, relationships, attributes and methods along with their visibilities.

3. Interaction diagrams- Sequence diagrams and Collaboration diagrams for different scenarios of the systems with all features like actors, objects and interactions.

4. Activity diagrams, State chart and other diagrams - Activity diagrams including the features like fork join and swim lanes. State diagrams including composite states and transitions. Component diagrams, Package diagrams and Deployment diagrams.
5. Forward and Reverser Engineering- Forward Engineering Class diagrams to classes in C++ and java and persistent classes to a database. Reverse Engineering C++ code, java code and a database.
6. Documentation using Rational Rose clear quest.

**Cycle II: Mini-Project** 8*3 periods= 24periods

The project deliverables include

- Problem statement
- Requirements Analysis

- Design
  - A Software Design Description and a System Design.
  - A test specification.
- Implementation
  - Implement the assigned project with one of the following web technologies
    - **Front end:** Java technologies/PHP/MS.NET Technologies
    - **Backend:** Oracle/My-SQL/SQL-Server
- Testing

**References :**


1. Project-based software engineering: An Object-oriented approach, Evelyn Stiller, Cathie LeBlanc, Pearson Education

2. Visual Modeling with Rational Rose 2002 and UML, Terry Quatrini, Pearson Education

| CSE 3.2.9 | WEB TECHNOLOGIES LAB | |
|---|---|---|
| **Instruction: 3 Periods/week,** | Univ. Exam: 3 Hours | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

Each student should develop two projects out of this list using JSP, JDBC, J2EE

1. Design Airlines Ticket Reservation System
2. Design ONLINE Banking system.
3. Design Library Information system
4. Design Gram Panchayat Information system for House tax, water tax, wealth tax, Library tax collection, phone bill, Electricity bill collection.
5. Design student information system portal which maintain attendance, marks etc.
6. Design online examination system.

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING(A) - VISAKHAPATNAM**
**I – SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION**
**Branch: COMPUTER SCIENCE AND ENGINEERING**
**IV/IV B.TECH** (FOUR YEAR COURSE) &
**IV/IV B.TECH** (SIX YEAR DOUBLE DEGREE COURSE)
(With effect from **2015-2016** admitted batch onwards)
Under Choice Based Credit System

**B.TECH. (CSE) 4th YEAR I-SEMESTER SCHEME OF INSTRUCTION AND EXAMINATION**

| SUB.REF. NO. | NAME OFTHE SUBJECT | PERIODS | | | MAXIMUM MARKS | | | CREDITS |
|---|---|---|---|---|---|---|---|---|
| | | THEORY | TUTORIAL | LAB | EXAM | SESSIONALS | TOTAL | |
| CSE 4.1.1 | EMBEDDED SYSTEMS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 4.1.2 | CYBER SECURITY & DIGITAL FORENSICS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 4.1.3 | ARTIFICIAL INTELLIGENCE | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 4.1.4 | PRINCIPLES OF ECONOMICS AND MANAGEMENT | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 4.1.5 | MOOCS-IV (UN-AUDIT) | -- | -- | -- | -- | 100 | 100 | -- |
| CSE 4.1.6 | BIGDATA ANALYTICS | 3 | 1 | -- | 70 | 30 | 100 | 4 |
| CSE 4.1.7 | KNOWLEDGE ENGINEERING LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 4.1.8 | BIGDATA ANALYTICS LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| CSE 4.1.9 | INTERNET OF THINGS LAB | -- | -- | 3 | 50 | 50 | 100 | 2 |
| | | | | | | **TOTAL CREDITS** | | 26 |

**OR**

| SUB.REF. NO. | NAME OF THE SUBJECT | MAXIMUM MARKS | | | CREDITS |
|---|---|---|---|---|---|
| | | INTERNAL | EXTERNAL | TOTAL | |
| **CSE 4.2** | **PROJECT/ THESIS WORK** | **50** | **50** | **100** | **15** |

| CSE 4.1.1 | EMBEDDED SYSYTEMS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To study the basics of embedded systems and its examples.
2. To study the 8051 Microcontroller architecture and its instruction set.
3. To discuss various software architectures in embedded systems.
4. To discuss Inter Task Communication procedures in RTOS and design issues of RTOS.
5. To study various embedded software development tools and debugging techniques.

**Course Outcomes:**

1. Student will be understand the basic architecture of 8051 micro controller.
2. ability to write ALP programs using 8051 instruction set.
3. Ability to understand the concepts related to RTOS and its Inter Task Communication methods.
4. Ability to understand various design issues of RTOS.
5. Understand about embedded software development tools.

**Syllabus:**

1. **Introduction to Embedded Systems:** Examples, Typical Hardware, Memory, Microprocessors , Busses; Introduction to 8051 Microcontroller , Architecture, Instruction set, Programming. **Interrupts**: Interrupt Basics, Shared-Data problem, Interrupt Latency.

2. **Software Architectures:** Round-Robin Architecture, Round-Robin with Interrupts Architecture, Function-Queue Scheduling Architecture, Real-Time Operating Systems Architecture, Selection of Architecture.

3. **Real Time Operating System**: Tasks and Task States, Tasks and Data, Semaphores and Shared Data, Semaphore Problems, Semaphore variants.

   **Inter Task Communication:** Message Queues, Mailboxes, Pipes, Timer Functions, Events, Memory Management, Interrupt Routines in RTOS Environment.

4. **Design issues of RTOS:** Principles , Encapsulation Semaphores and Queues, Hard Real-Time Scheduling Considerations, Saving Memory Space, Saving Power.

5. **Embedded Software development Tools:** Host and Target Machines , Linker/Locator for Embedded Software, Getting Embedded Software into the Target System.

   **Embedded Software Debugging Techniques :**Testing on your Host Machine, Instruction Set Simulators, Laboratory Tools used for Debugging.

6. **Introduction to the Internet of Things:** History of IoT, IoT Architecture, M2M – Machine to Machine, Web of Things, IoT protocols, The Layering concepts, IoT Communication Pattern, IoT protocol Architecture.

**Text Books:**

1. The 8051 Microcontroller Architecture, Programming & Applications, Kenneth J. Ayala, Penram International.
2. An Embedded Software Primer, David E. Simon, Pearson Education , 2005.
3. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Marina Ruggieri & Homayoun Nikookar, River Publishers Series in Communications.

**Reference Book:**

1. Embedded Systems: Architecture , Programming and Design, Raj Kamal, Tata McGraw-Hill Education, 2008

| CSE 4.1.2 | CYBER SECURITY & DIGITAL FORENSICS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**1. Introduction to Information Security Fundamentals and Best Practices:** Protecting Your Computer and its Contents, Securing Computer Networks--Basics of Networking, Compromised Computers, Secure Communications and Information Security Best Practices, Privacy Guidelines, Safe Internet Usage.

**2. Ethics in Cyber Security & Cyber Law:** Privacy, Intellectual Property, Professional Ethics, Freedom of Speech, Fair User and Ethical Hacking, Trademarks, Internet Fraud, Electronic Evidence, Cybercrimes.

**3. Penetration Testing:** Overview of the web from a penetration testers perspective, Exploring the various servers and clients, Discussion of the various web architectures, Discussion of the different types of vulnerabilities, Defining a web application test scope and process, Defining types of penetration testing.

4. **Web Application Security:** Common Issues in Web Apps, What is XSS, SQL injection, CSRF, Password Vulnerabilities, SSL, CAPTCHA, Session Hijacking, Local and Remote File Inclusion, Audit Trails, Web Server Issues.

**5. Forensics & Network Assurance:** Forensic Technologies, Digital Evidence Collection, Evidentiary Reporting, Layered Defense, Surveillance and Reconnaissance, Outsider Thread Protection

6. **Information Risk Management:** Asset Evaluation and Business Impact Analysis, Risk Identification, Risk Quantification, Risk Response Development and Control, Security Policy, Compliance, and Business Continuity. Forensic investigation using Access Data FTK, En-Case

**7. Cyber Incident Analysis and Response:** Incident Preparation, Incident Detection and Analysis. Containment, Eradication, and Recovery. Proactive and Post-Incident Cyber Services, CIA triangle

Books:

1. The Official CHFI Study Guide for Computer Hacking Forensic Investigator by Dave Kleiman
2. CISSP Study Guide, 6th Edition by James M. Stewart
3. www.**nist**.gov/

| CSE 4.1.3 | ARTIFICIAL INTELLIGENCE | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To learn about AI problem, Production Systems and their characteristics.
2. To understand the importance of search and the corresponding search strategies for solving AI problem.
3. To introduce to Planning, Natural Language Processing and Expert Systems.

**Course Outcomes:**

1. The Student understands AI problem characteristics, state space approach for solving AI problem, Production System framework.
2. The student learn several optimal search strategies and the use of heuristics.
3. The student learns relational, inferential, inheritable and procedural knowledge and the corresponding knowledge representation approaches.
4. The student is introduced to applying AI problem solving approaches to natural language processing, planning and expert systems.

**Syllabus:**

1. **Introduction to Artificial Intelligence:** Artificial Intelligence, AI Problems, AI Techniques, Defining the Problem as a State Space Search, Problem Characteristics, Production Systems.

2. **Search Techniques:** Issues in The Design of Search Programs, Un-Informed Search, BFS, DFS; Heuristic Search Techniques: Generate-And- Test, Hill Climbing, Best-First Search, $A^*$ Algorithm, Problem Reduction, $AO^*$ Algorithm, Constraint Satisfaction, Means-Ends Analysis.

3. **Knowledge Representation using Rules:** Procedural Vs Declarative Knowledge, Logic programming, Forward Vs Backward Reasoning, Matching Techniques, Partial Matching, RETE Matching Algorithm AI Programming languages: Overview of LISP and PROLOG, Production System in Prolog.

4. **Symbolic Logic**: Propositional Logic, First Order Predicate Logic: Representing Instance and is-a Relationships, Computable Functions and Predicates, Unification & Resolution, Natural Deduction; **Structured Representations of Knowledge**: Semantic Nets, Partitioned Semantic Nets, Frames, Conceptual Dependency, Conceptual Graphs, Scripts.

5. **Reasoning under Uncertainty**: Introduction to Non-Monotonic Reasoning, Truth Maintenance Systems, Logics for Non-Monotonic Reasoning, **Statistical Reasoning**: Bayes Theorem, Certainty Factors and Rule-Based Systems, Bayesian Probabilistic Inference, Bayesian Networks, Dempster- Shafer Theory, Fuzzy Logic: Crisp Sets ,Fuzzy Sets, Fuzzy Logic Control, Fuzzy Inferences & Fuzzy Systems.

6. **Natural Language Processing**: Steps in The Natural Language Processing, Syntactic Processing and Augmented Transition Nets, Semantic Analysis, NLP Understanding

Systems; **Planning:** Components of a Planning System, Goal Stack Planning, Non-linear Planning using Constrait Posting, Hierarchical Planning, Reactive Systems.

7. **Experts Systems:** Overview of an Expert System, Architecture of an Expert Systems, Different Types of Expert Systems- Rule Based, Frame Based, Decision Tree based, Case Based, Neural Network based, Black Board Architectures, Knowledge Acquisition and Validation Techniques, , Knowledge System Building Tools, Expert System Shells.

**Text Book:**


1. Artificial Intelligence, Elaine Rich and Kevin Knight, Tata Mcgraw-Hill Publications
2. Introduction To Artificial Intelligence & Expert Systems, Patterson, PHI publications


**References:**


1. Artificial Intelligence, George F Luger, Pearson Education Publications
2. Artificial Intelligence : A modern Approach, Russell and Norvig, Printice Hall
3. Artificial Intelligence, Robert Schalkoff, Mcgraw-Hill Publications
4. Artificial Intelligence and Machine Learning, Vinod Chandra S.S., Anand Hareendran S.

| CSE 4.1.4 | PRINCIPLES OF ECONOMICS & MANAGEMENT | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. Apply economic reasoning to the analysis of selected contemporary economic problems.
2. Understand how households (demand) and businesses (supply) interact in various market structures to determine price and quantity of goods and services produced and consumed.
3. Analyze the efficiency and equity implications of government interference in markets.
4. Recognize and identify situations leading to market failures and government failures.
5. Evaluate the intent and outcomes of government stabilization policies designed to correct macroeconomic problems.
6. Use economic problem solving skills to discuss the opportunities and challenges of the increasing globalization of the world economy.

**Course Outcomes:**

1. Understand the links between production costs and the economic models of supply.
2. Represent supply, in graphical form, including the upward slope of the supply curve and what shifts the supply curve.
3. Understand the efficiency and equity implications of market interference, including government policy.
4. Understand how different degrees of competition in a market affect pricing and output.
5. Apply economic reasoning to individual and firm behavior.

**Syllabus:**

1. **Introduction to Managerial Economics:** Wealth, Welfare and Scarce Definitions of Economics; micro and Macro Economics; Demand- Law of Demand, Elasticity of Demand, types of Elasticity and factors of determining price elasticity of Demand: utility- Law of Diminishing Marginal Utility and its limitations.

2. **Conditions of Different Market Structures:** Perfect Competition, Monopolistic Competition, Monopoly, Oligopoly, and Duopoly.

3. **Forms of Business Organizations:** Sole Proprietorship, Partnership, Joint Stock Company- Private Limited and Public Limited Companies, Public Enterprises and their types.

4. **Introduction to Management:** Functions of Management- Taylor's Scientific management; Henry Fayol's Principle of Management; Human Resource Management-basic Functions of HR Manager; Man Power Planning, Recruitment, Selection, Training, Development, Placement, Compensation and performance Appraisal( in brief).

5. **Production Management:** Production Planning and Control, plant Location, Break-Even Analysis, assumptions and applications.

6. **Financial Management:** Types of Capital: Fixed and Working Capital , and Methods of Raining Finance; Depreciation: Straight Line and Diminishing Balance Methods. Marketing Management: Functions of marketing and Distribution Channels.

7. **Entrepreneurship:** Entrepreneurial Functions, Entrepreneurial Development: Objectives, Training, Benefits: Phases of Installing a project

**Text Books:**


1. K.K.DEWETT, **Modern Economic Theory**, S.Chand and Company, New Delhi-55.
2. S.C. Sharma and Banga T. R., Industrial Organization & Engineering Economics, khanna Publications, Delhi-6.


**Reference Books:**


1. A.R. AryaSri, Management Science, TMH publications, New Delhi-20.
2. A.R. AryaSri, Managerial Economics and Financial Analysis, TMH Publications, new Delhi-20.

| CSE 4.1.6 | BIGDATA ANALYTICS | |
|---|---|---|
| **Instruction: 3 Periods + 1 Tut/week,** Univ. Exam: 3 Hours | | **Credits: 4** |
| **Internal: 30 Marks** | **University Exam: 70 Marks** | **Total: 100 Marks** |

**Course Objectives:**

On completing this course student will be able to

1. Understand big data and Apache Hadoop Eco system
2. Understand distributed , parallel, cloud computing and SQL concepts
3. Apply Hadoop concepts
4. Understand concepts of map and reduce and functional programming

**Course Outcomes :**

1. Gain conceptual understanding of analytics concepts, algorithms and statistical tests
2. Students will be able to look at the core projects used for both batch and real time data processing such as Hadoop
3. Students will be able to look at wider range of problems and data science based solutions

**Syllabus:**

1. **Introduction to Big Data:** Big Data-definition, Characteristics of Big Data (Volume, Variety, Velocity, Veracity, Validity), Importance of Big Data , Patterns for Big Data Development, Data in the Warehouse and Data in Hadoop,

2. **Introduction to Hadoop:** Hadoop- definition, Understanding distributed systems and Hadoop, Comparing SQL databases and Hadoop, Understanding MapReduce, Counting words with Hadoop—running your first program, History of Hadoop, Starting Hadoop - The building blocks of Hadoop, NameNode, DataNode, Secondary NameNode, JobTracker and Task Tracker

3. **MapReduce** -A Weather Dataset, Analyzing the Data with Unix Tools, Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes, Developing a MapReduce Application - The Configuration API, Configuring the Development Environment, Running Locally on Test Data, Running on a Cluster, Tuning a Job, MapReduce Workflows

4. **HDFS:** Components of Hadoop -Working with files in HDFS, Anatomy of a MapReduce program, Reading and writing the Hadoop Distributed File system -The Design of HDFS, HDFS Concepts, The Command-Line Interface, Hadoop Filesystem, The Java Interface, Data Flow, Parallel Copying with distcp, Hadoop Archives

5. **MapReduce Programming:** Writing basic Map Reduce programs - Getting the patent data set, constructing the basic template of a Map Reduce program, Counting things, Adapting for Hadoop's API changes, Streaming in Hadoop, Improving performance with combiners.

6. **MapReduce Advanced Programming:** Advanced MapReduce - Chaining MapReduce

jobs, joining data from different sources, creating a Bloom filter, Passing job-specific parameters to your tasks, probing for task-specific information, Partitioning into multiple output files, Inputting from and outputting to a database, keeping all output in sorted order

**7. Graph Representation in MapReduce:** Modeling data and solving problems with graphs, Shortest Path Algorithm, Friends-of-Friends Algorithm, PageRank Algorithm, Bloom Filter, Parallelized Bloom filter creation in MapReduce, Map-Reduce semi-join with Bloom filters

**Textbooks**:

1. Dirk deRoos, Chris Eaton, George Lapis, Paul Zikopoulos, Tom Deutsch "Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data", 1st Edition, TMH,2012.
2. Hadoop: The Definitive Guide by Tom White, 3rd Edition, O'reilly

**Reference Books:**

1. 1.Hadoop in Action by Chuck Lam, MANNING Publ.
2. Hadoop in Practice by Alex Holmes, MANNING Publishers
3. Mining of massive datasets, Anand Rajaraman, Jeffrey D Ullman, Wiley Publications.

| CSE 4.1.7 | KNOWLEDGE ENGINEERING LAB | |
|---|---|---|
| **Instruction: 3 Periods** | Univ. Exam: 3 Hours | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**Course Objectives:**

1. To study the various data analysis techniques in R Programming language.
2. To apply the various data mining techniques available inWEKA for generating Knowledge such as Association Analysis, Classification and Clustering to various standard datasets and own datasets.
3. To build Rule based Knowledge Systems using forward chaining and Backward chaining using CLIPS and PROLOG respectively

**Course Outcomes:**

1. Student will be able to write R programs to perform several data analytics operations on datasets
2. Ability to extract patterns by applying appropriate data mining techniques from different types of datasets using WEKA.
3. Ability to apply knowledge represented in the form of rules to draw conclusions using either forward or backward chaining using CLIPS /PROLOG.

## 1. Exploratory data analysis using R

1. Load the 'iris. CSV' file and display the names and type of each column. Find statistics such as min, max, range, mean, median, variance, standard deviation for each column of data.
2. Write R program to normalize the variables into 0 to 1 scale using min-max normalisation
3. Generate histograms for any one variable (sepal length/ sepal width/ petal length/ petal width) and generate scatter plots for every pair of variables showing each species in different color .
4. Generate box plots for each of the numerical attributes. Identify the attribute with the highest variance.
5. Study of homogeneous and heterogeneous data structures such as vector, matrix, array, list, data frame in R.
6. Write R Program using 'apply' group of functions to create and apply normalization function on each of the numeric variables/columns of iris dataset to transform them into a value around 0 with z-score normalization.

7. a) Use R to apply linear regression to predict evaporation coefficient in terms of air velocity using the data given below:

| Air Velocity (cm/sec) | 20,60,100,140,180,220,260,300,340,380 |
|---|---|
| Evaporation Coefficient (sqmm/sec) | 0.18, 0.37, 0.35, 0.78, 0.56, 0.75, 1.18, 1.36, 1.17, 1.65 |

   b) Analyze the significance of residual standard-error value, R-squared value, F-statistic. Find the correlation coefficient for this data and analyze the significance of the correlation value.
   c) Perform a log transformation on the 'Air Velocity 'column, perform linear regression again, and analyze all the relevant values.
8. Write R Program using 'apply' group of functions to create and apply normalization function on each of the numeric variables/columns of iris dataset to transform them a value around 0 with z-score normalization.

## 2.WEKA Knowledge Extraction toolkit:

9. Create an ARFF (Attribute-Relation File Format) file and read it in WEKA. Explore the purpose of each button under the preprocess panel after loading the ARFF file. Also, try to interpret using a different ARFF file, *weather.arff*, provided with WEKA**.**

10. Performing data preprocessing in Weka
Study **Unsupervised Attribute Filters** such as ***Replace MissingValues*** to replace missing values in the given dataset, ***Add*** to add the new attribute *Average,* ***Discretize*** to discretize the attributes into bins. Explore ***Normalize*** and ***Standardize*** options on a dataset with numerical attributes.

11. Classification using the WEKA toolkit
Demonstration of classification process using id3 algorithm on categorical dataset(weather).
Demonstration of classification process using naïve Bayes algorithm on categorical dataset ('vote').
Demonstration of classification process using Random Forest algorithm on datasets containing large number of attributes.

12.Classification using the WEKA toolkit – Part 2
Demonstration of classification process using J48 algorithm on mixed type of dataset after discretizing numeric attributes.
Perform cross-validation strategy with various fold levels. Compare the accuracy of the results.

13.Performing clustering in WEKA
Apply hierarchical clustering algorithm on numeric dataset and estimate cluster quality. Apply DBSCAN algorithm on numeric dataset and estimate cluster quality.

14. Association rule analysis in WEKA
Demonstration of Association Rule Mining on supermarket dataset using Apriori Algorithm with different support and confidence thresholds.
Demonstration of Association Rule Mining on supermarket dataset using FP-Growth Algorithm with different support and confidence thresholds.

## 3. Building Knowledge based Inference Systems:

15. Implement AI problem solving through Rule based forward chaining inference using public domain software tool like CLIPS.
16. Implement AI problem solving through Rule based Backward chaining inference using PROLOG

**References:**

Practical data science with R, Nina Zumel and John Mount- Dreamtech Press.

| CSE 4.1.8 | BIG DATA ANALYTICS LAB | |
|---|---|---|
| **Instruction: 3 Periods** | Univ. Exam: 3 Hours | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**1. Getting Hadoop Up and Running in a cluster:**

1. Setting up Hadoop on standalone machine.
2. Wordcount Map Reduce program using standalone Hadoop.
3. Adding the combiner step to the Wordcount Map Reduce program.
4. Setting up HDFS.
5. Using HDFS monitoring UI
6. HDFS basic command-line file operations.
7. Setting Hadoop in a distributed cluster environment.
8. Running the WordCount program in a distributed cluster environment.
9. Using Map Reduce monitoring UI

**2. Hadoop Map Reduce Applications:**

10. Choosing appropriate Hadoop data types.
11. Implementing a custom Hadoop Writable data type.
12. Implementing a custom Hadoop key type.
13. Emitting data of different value types from a mapper.
14. Choosing a suitable Hadoop Input Format for your input data format.
15. Formatting the results of Map Reduce Computation – using Hadoop Output Formats.

**3. Analytics**

16. Simple analytics using Map Reduce.
17. Performing Group-By using Map Reduce.
18. Calculating frequency distributions and sorting using Map Reduce.
19. Plotting the Hadoop results using GNU plot.
20. Calculating histograms using Map Reduce.
21. Calculating scatter plots using Map Reduce.
22. Parsing a Complex dataset with Hadoop.
23. Joining two datasets using Map Reduce.

**Text Book:**

1. Hadoop Map Reduce Cookbook, Srinath Perera & Thilina Gunarathne, 2013, PACKT PUBLISHING.

| CSE 4.1.9 | INTERNET OF THINGS LAB | |
|---|---|---|
| **Instruction: 3 Periods** | Univ. Exam: 3 Hours | **Credits: 2** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**LIST OF EXPERIMENTS:**

1. STUDY OF VARIOUS NETWORK PROTOCOLS USED IN IOT.
2. APPLICATION OF WIFI IN IOT SYSTEMS.
3. APPLICATION OF 6LOWPAN IN IOT SYSTEMS.
4. APPLICATION OF BLUETOOTH IN IOT SYSTEMS.
5. APPLICATION OF 802.15.4 ZIGBEE. IN IOT SYSTEMS.
6. DESIGN A SIMPLE IOT SYSTEM COMPRISING SENSORS, WIRELESS NETWORK CONNECTION, DATA ANALYTICS

| CSE 4.2 | PROJECT / THESIS WORK | |
|---|---|---|
| **Instruction: 6 Periods/week** | Hours | **Credits: 14** |
| **Internal: 50 Marks** | **University Exam: 50 Marks** | **Total: 100 Marks** |

**GUIDELINES for doing the Project Work:**

1. Candidates can do their thesis work within the department or in any industry/research organization for one semester in the 4th year of their study. In case of project done in an industry/research organization, one advisor (Guide) should be from the department and one advisor (Co-Guide) should be from the industry/research organization.
2. Students should work in teams of 3 to 4 members and submit thesis on the project work done by them.

**Format For Preparation Of Project Thesis For B. Tech (CSE):**

1. Arrangement Of Contents:

The sequence in which the project report material should be arranged and bound should be as follows:

1. Cover Page & Title Page
2. Bonafide Certificate
3. Abstract
4. Table of Contents
5. List of Tables
6. List of Figures
7. List of Symbols, Abbreviations and Nomenclature
8. Chapters
9. Appendices
10. References

*The table and figures shall be introduced in the appropriate places.