Scheme of Instruction and Syllabi
of
Choice Based Credit System (CBCS) of

BE / B.TECH V AND VI SEMESTERS
OF
FOUR YEAR DEGREE COURSE
IN

COMPUTER SCIENCE & ENGINEERING

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY™
(An Autonomous Institution)
Affiliated to OU; All U.G. and 5 P.G. Programmes (Civil, CSE, ECE, Mech. & EEE)
Accredited by NBA; Accredited by NAAC - 'A' Grade (UGC); ISO Certified 9001:2015
Chaitanya Bharathi P.O, CBIT Campus, Gandipet, Kokapet (V),
Gandipet Mandal, Ranga Reddy District, Hyderabad-500075, Telangana
email: principal@cbit.ac.in; Website: www.cbit.ac.in Ph : 040-24193276 / 277 / 279

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
SCHEME OF INSTRUCTION AND EXAMINATION
V-Semester of B.E under CBCS
COMPUTER SCIENCE AND ENGINEERING

SEMESTER-V

Sl.No	Course Code	Title of the Course	Scheme of Instruction		Scheme of Examination			Credits
			Hours per week		Duration of SEE in Hours	Maximum Marks		
			L/T	P/D		CIE	SEE	
THEORY								
1	16CSC17	Design and Analysis of Algorithms	3/1	-	3	30	70	4
2	16CSC18	Automata Languages and Computation	3/1	-	3	30	70	4
3	16CSC19	Operating Systems	3	-	3	30	70	3
4	16CSC20	Data Communication and Computer Networks	3	-	3	30	70	3
5	16CSC21	Software Engineering	3	-	3	30	70	3
6	16CSE 04/05/06	Elective - II	3	-	3	30	70	3
PRACTICALS								
7	16CSC22	Operating Systems Lab	-	3	3	25	50	2
8	16CSC23	Data Communication and Computer Networks Lab	-	3	3	25	50	2
9	16CSC24	Software Engineering Lab	-	3	3	25	50	2
TOTAL			20	9	-	255	570	26

Elective-II:

16CSE 04 - Mobile Application Development
16CSE 05 - Computer Graphics
16CSE 06 - Advanced Computer Architecture

L: Lecture T: Tutorial D: Drawing P: Practical
CIE - Continuous Internal Evaluation SEE - Semester End Examination

Assessment Procedure				
Course (in terms of credits)	**Continuous Internal Evaluation (Marks)**	**Semester end Examination (Marks)**	**Remarks**	**Duration of Semester End Examination**
Three(3)Credits/ Four(4)credits	30*	70**	Theory Course/ Engg . Graphics	3 Hours
Two(2) Credits	25	50	Lab Course/Workshop	3 Hours
One(1) Credit	15	35	Lab Course	2 Hours
One(1) Credit	50	-	Mini Project	-

* Out of 30 CIE, 10 marks are allotted for slip-tests (Three slip tests will be conducted, each of ten marks, and average of best two is considered) and the remaining 20 marks are based on the average of two tests, weightage for each test is 20 marks.

** The question paper will be in two parts, Part-A and Part-B. Part A is compulsory and contains short answer questions covering the entire syllabus, and carries 20 marks. Part-B carries 50 marks and covers all the units of the syllabus (student has to answer five out of seven questions).

Note: A course that has CIE but no SEE as per scheme is treated as PASS/FAIL for which pass marks are 50% of CIE.

A candidate has earned the credits of a particular course, if he/she secures not less than the minimum marks/ grade as prescribed. Minimum pass marks in the SEE plus CIE shall be 40% for theory courses/subjects and 50% for lab courses /Mini Project Project.

16CSC17

DESIGN AND ANALYSIS OF ALGORITHMS

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To provide an introduction to formalisms to understand, analyze and denote time complexities of algorithms.
2. To introduce the different algorithmic approaches for problem solving through numerous example problems.
3. To provide some theoretical grounding in terms of finding the lower bounds of algorithms and the NP-completeness.

Course Outcomes:

1. Describe asymptotic notation used for denoting performance of algorithms.
2. Analyze the performance of a given algorithm and denote its time complexity using the asymptotic notation for recursive and non-recursive algorithms.
3. List and describe various algorithmic approaches.
4. Solve problems using divide and conquer, greedy, dynamic programming, backtracking and branch and bound algorithmic approaches.
5. Apply graph search algorithms to real world problems.
6. Demonstrate an understanding of NP- Completeness theory and lower bound theory

UNIT-I

Introduction: Notation for Algorithm Specification, Insertion sort specification and analysis

Growth of functions: Asymptotic notation, standard notation and common functions.

Recurrences: The substitution method, the recursion-tree method, the Master method

Set representation: Simple UNION and FIND, Weighted Union and collapsing Find.

UNIT-II

Divide-and Conquer: The general method, specification and analysis of: finding maximum minimum of a set of values, quick sort, merge sort, Strassen's Matrix multiplication.

Greedy Method: The general method, Knapsack problem, Optimal Storage on tapes, Job sequencing with deadlines, Optimal merge patterns, Huffman codes.

UNIT-III

Dynamic Programming: The general method, Multistage graph, Floyd-Warshall algorithm, Bellman-Ford algorithm, Optimal Binary Search trees, 0/1 Knapsack, Traveling Salesman Problem, Matrix-Chain multiplication and Longest Common Subsequence.

UNIT-IV

Backtracking: The general method,8-Queens Problem, Sum of subsets, Graph Coloring, Hamiltonian cycle, 0/1 Knapsack Problem

Branch and Bound: The general method, Least cost search, control abstraction for LC-Search, Bounding, FIFO branch and bound,LC branch and bound, 0/1 knapsack problem, Traveling salesperson problem.

Depth first Search: Bi-connected components, topological sorting, strongly connected components.

UNIT-V

Lower Bound Theory: Comparison trees for searching and sorting

NP-Completeness: Basic concepts, Polynomial time, polynomial time verification, reducibility

NP-complete problems: The clique problem, the vertex-cover problem, the Hamiltonian cycle problem, the traveling salesman problem and the subset sum problem.

Text Books:

1. Horowitz E. Sahani S:"Fundamentals of Computer Algorithms", 2nd Edition, Galgotia Publications, 2008.
2. Cormen, Leiserson, Rivest,Stein: "Introduction to Algorithms", 3rdEdition, PHI Learning, 2017.
3. Aho, Hopcroft, Ulman, "The Design and Analysis of Computer Algorithms", Pearson Education, 2000.

Online Resources:

1. http://nptel.ac.in/courses/106101060/

16CSC18

AUTOMATA LANGUAGES AND COMPUTATION

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To introduce the students to the theoretical concepts of computer science.
2. To know the various languages and grammars that are associated with various recognizers.
3. To understand the language by considering the idea of a decision problem
4. To understand language recognition problem and different classes of a problem.

Course Outcomes:

1. Demonstrate a fundamental understanding of the core concepts in automata theory and formal languages.
2. Design grammars and automata (recognizers) for different language classes.
3. Identify formal language classes and prove language membership properties.
4. Prove and disprove theorems establishing key properties of formal languages and automata.
5. Demonstrate an understanding of the principles behind the basic abstract computing model and its variants.

UNIT-I

Automata: Introduction to Chomsky's Hierarchy, The need to study automata theory, Central Concepts of Automata Theory.

Finite Automata: An Informal Picture of Finite Automata, Deterministic Finite Automata, Non-deterministic Finite Automata, Finite automata for text search, Finite Automata with Epsilon Transitions.

UNIT-II

Regular expressions & Languages: Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions.

Properties of Regular Languages: Pumping Lemma for regular languages, Closure properties of Regular Languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT-III

Context Free Grammars and Languages: Context free grammars, Parse Trees, Right Linear and Left Linear Grammars, Applications of CFGs, Ambiguity in Grammars and Languages.

Pushdown Automata: Definition of the Pushdown Automaton, Languages of PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.

UNIT-IV

Properties of Context Free Languages: Normal Forms for Context Free Grammars, Pumping Lemma, Closure Properties of CFLs, Decision Properties of CFLs. LR(0) grammars, LR(0) and DPDA, LR(k) grammars.

UNIT-V

Introduction to Turing Machines: Problems that Computers cannot Solve, The Turing machines, Programming Techniques for Turing Machines, Extensions to the basic Turing Machine, Restricted Turing Machines, Turing machines and Computers.

Un-decidability: A language that is not Recursively Enumerable, An undecidable problem that is recursively enumerable, Undecidable problems about Turing Machines, Post's Correspondence Problem, Other Undecidable Problems.

.

Text Book:

1. John. E. Hopcroft, Rajeev Motwani, Jeffery, D. Ullman, "Introduction to Automata Theory, Languages and Computation", 3rd edition, Pearson Education, 2008.

Suggested Readings:

1. John C.Martin, "Introduction to Languages and the Theory of Computation", 3rd edition Tata McGraw Hill, 2007.
2. Mishra and Chandrashekaran, "Theory of Computer Science – Automata languages and computation", 3rd edition, PHI,2008.

16CSC19

OPERATING SYSTEMS

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To understand the services an operating system provides to users, processes and other systems.
2. To understand how to manage various resources like CPU, Memory, Files and I/O.
3. To understand Process Synchronization, multiprogramming, Deadlocks.
4. To understand the Architecture and implementation of different operating systems.

Course Outcomes:

1. To develop the knowledge of the role of operating system and its design.
2. To implement the knowledge of multiprogramming, multithreading, deadlocks.
3. To analyze the concept of IPC and resource sharing among the users.
4. To understand memory management including virtual memory.
5. Analyze various Disk scheduling algorithms and I/O operation implementation techniques
6. Familiar with security mechanisms and understand the features of Linux and Windows Operating systems

UNIT-I

Introduction: Definition, Operating System Structure, Operating System Services, System Calls, System programs, Operating System Design and Implementation.

Processes & Threads: Process concept, Process Scheduling, Inter-process communication, Threads, Multithreading Models.

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiprocessor scheduling.

UNIT-II

Memory Management: Swapping, Contiguous allocation, Paging, Segmentation, Segmentation with Paging.

Virtual memory: Demand paging, Page replacement Algorithms, Allocation of Frames, Thrashing.

File System Interface: File Concept, Access Methods, Directory and Disk Structure, File System Mounting.
File System Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free Space management.

UNIT-III
Process synchronization: Critical Section problem, Synchronization Hardware, Semaphores, Classical problems of Synchronization, Monitors
Deadlocks: System model, Deadlock Characterization, Methods for handling deadlocks, Prevention, Avoidance, Detection, Recovery from Deadlock.

UNIT-IV
Mass-Storage Structure: Overview of Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure, Stable-Storage Implementation.
I/O System: I/O hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O request to hardware operation, Streams, Performance.

UNIT-V
Protection: Goals of Protection, Domain of protection, Access matrix, Implementation of Access matrix.
Security: The Security Problem, Program Threats, System and Network Threats, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Networks, Computer-Security Classifications
Case Studies: Linux System: Design Principles, Kernel Modules, Process Management, Network Structure, And Security. Windows - Design Principles, Architecture, Environmental Subsystem.

Text Books:

1. AviSilberchatz, Peter B. Galvin, Greg Gagne, "Operating System-Concepts", John Wiley & sons, 9th Edition, 2016

Suggested Reading:

1. Andrew S. Tanenbaum, "Modern Operating Systems", 2nd Edition (2001), Pearson Education, Asia
2. W. Richard Stevens; Stephen A. Rago, "Advanced Programming in the UNIX Environment", Third Edition, Addison-Wesley professional Publication Date:14-MAY-2013
3. Dhananjay, Dhamdhere.M, Operating System-concept based approach, 3rd edition (2009), Tata McGraw Hill, Asia

16CSC20

DATA COMMUNICATION AND COMPUTER NETWORKS

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. Understanding the concepts of data communications.
2. Understanding the concepts of network reference models.
3. Analysis of routing algorithms and congestion control algorithms.
4. Functionality of the transport layer.
5. Understand different application layer protocols.

Course Outcomes:

1. Understand the communication protocol suites like ISO-OSI and TCP/IP.
2. Understand and explain Data Communications System and its components
3. Identify and evaluate various routing algorithms, congestion control algorithms.
4. Identify and use internet protocols like IP, ARP, ICMP, IGMP, BGP, OSPF, and DHCP etc.
5. Know the working of transport layer protocols like TCP, UDP, RTCP etc.
6. Understand about the applications (like WWW, DNS, email etc.) and the underlying protocols.

UNIT-I

Introduction: Data Communication, Network Types, Network Models – Protocol Layering, TCP/IP Protocol Suite, OSI Model, OSI vs TCP/IP.
Physical Layer: Transmission Media (wired and wireless), Switching.

UNIT-II

Data Link Layer:DLL design issues, Error detection and correction, elementary data link protocols, sliding window protocols, Multiple access protocols.
LAN: Wired LAN, Wireless LAN, Connecting devices and Wireless LAN.

UNIT-III

Network Layer: Network layer design issues, Routing algorithms, congestion control algorithms, Quality of service, Internetworking, Network layer in the internet.

UNIT-IV

Transport Layer : Elements of transport protocol, congestion control, TCP, UDP.

UNIT-V

Application Layer :WWW and HTTP, FTP, Email, TELNET, SSH, DNS

Multimedia : Compression, Multimedia data, Multimedia in the internet, Real-time interactive protocols.

Text Books:

1. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw– Hill, Fifth Edition, 2013.
2. A. S. Tanenbaum, "Computer Networks", Pearson Education, Fifth Edition, 2013
3. William Stallings, "Data and Computer Communication", Eighth Edition, Pearson Education, 2007.

Suggested Reading:

1. Larry L.Peterson, Peter S. Davie, "Computer Networks", Elsevier, Fifth Edition, 2012.
2. James F. Kurose, Keith W. Ross, "Computer Networking: A Top–Down Approach Featuring the Internet", Pearson Education, 2005.

16CSC21

SOFTWARE ENGINEERING PRINCIPLES AND PRACTICES

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To Understand the software Engineering Practice & Process Models.
2. To understand Design Engineering and Software Project management.
3. To gain knowledge of the overall project activities.

Course Outcomes :

1. Assessment in each module gives the overall Software engineering practice.
2. Ability to enhance the software project management skills.
3. Ability to comprehend the systematicmethodologiesinvolved in SE.
4. Ability to design and develop a software product in accordance with SE principles.

UNIT-I

Introduction to Software Engineering: The nature of Software, Software Engineering, The Software Process, software Engineering Practice.

Process Models: A Generic Process Model, Process Framework, CMMI, Prescriptive Process Models: Waterfall Model, Incremental Process Models, Evolutionary Process Models – Prototyping, The Spiral Model, ConcurrentModels; **An agile view of Process:** Agility, Agile Process and Agile Process Models –Extreme Programming (XP), Adaptive Software Development(ASD).

UNIT-II

Requirement Engineering – Understanding Requirements : Establishing the Groundwork, Requirement Engineering tasks, Initiating the Requirements Engineering Process, Eliciting Requirements, Feasibility Study,

Software RequirementsAnalysis and Specification: Software Requirements, Problem Analysis, Requirements Specification, Decision Tables, SRS Document, IEEE standards for SRS, Case Studies.

Planning and Managing the project: Managing Software Project, Project Personnel, Effort Estimation, Risk Management, the project plan, Software project estimation – Empirical estimation models.

UNIT-III

Design Engineering: Design Principles, Design Notation and Specification, Design concepts, Flow oriented modeling; The function-oriented design for the case studies; OO Design Concepts; Modeling Component-Level Design.

Architectural Design: Software Architecture, Data Design, A brief Taxonomy of Architectural Styles. **Implementation:** Coding Principles and Standards, Coding Process, Code Verification.

UNIT-IV

Testing Strategies: A strategic approach to software testing, strategic issues, test strategies for Conventional and OO Software, Validation Testing, System Testing, Art of Debugging.

Testing Tactics: Software Testing Fundamentals, White Box Testing: Basis Path Testing, Control Structure Testing, O-O Testing methods, Black Box, Software quality.

UNIT-V

Software Quality Assurance – Managing Software Project, Quality concepts, Software Quality Assurance Software Reviews, Technical Reviews, Software reliability;

Software Configuration Management: Identification of Objects in the Software Configuration, Configuration Audit, SCM Standards

Software Maintenance: Categories of Maintenance, Maintenance Process models, Software reuse, Metrics for maintenance.

Text Books:

1. Roger S. Pressman, "Software Engineering: A practitioner's approach", McGraw Hill,
2. Shari Lawrence Pfleeger, "Software Engineering Theory and Practices", 4th Edition, Pearson Education, India, 2011.
3. Pankaj Jalote"An integrated approach to Software Engineering", Springer/ Narosa, 2014

Suggested Reading:

4. Sommerveli "Software Engineering", 10TH Edition, Pearson,2015
5. Rajib Mal "Fundamental of Software Engineering", 4th Edition, PHI Learning, 2014.

16CSE04

MOBILE APPLICATION DEVELOPMENT

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives

1. Understand J2ME and Android architecture and solve problems with J2ME, Android application.
2. Design, implement and evaluate a User Interface for a mobile application.
3. Understand how to create working mobile application for small computing devices using Android.
4. Understand to manage repository of data information for mobile application
5. Categorize the challenges posed by developing mobile applications and able to propose and evaluate and select appropriate solutions.

Course Outcomes

1. Ability to evaluate and select appropriate solutions to the mobile computing platform.
2. Ability to develop the user interface.
3. Ability to develop database management system to retrieve data for mobile application.
4. Ability to build a simple mobile application.
5. Develop and Deploy mobile applications.

UNIT- I

Developing for Mobile and Embedded Devices, J2ME Overview: Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices.

J2ME Architecture and Development Environment: J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, J2ME Software Development Kits, Multiple MlDlet in a MIDlet Suite.

UNIT-II

Commands, Items, and Event Processing: J2ME User Interfaces, Display Class, Command Class, Item Class, Exception Handling.

Record Management System: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener.

UNIT- III

Generic Connection Framework: The Connection, Hypertext Transfer Protocol, Communication Management Using HTTP Commands, Session Management, Transmit as a Background Process

Android: An Open Platform for Mobile Development, A Little Background, Native Android Applications, Android SDK Features, Developing for Android, Android Development Tools

UNIT- IV

Creating Applications and Activities:Introducing the Application Manifest File, Externalizing Resources, The Android Application Lifecycle, A Closer Look at Android Activities,

Building User Interfaces:Fundamental Android UI Design, Android User Interface Fundamentals, Introducing Layouts

UNIT- V

Databases and Content Providers: Introducing Android Databases, Working with SQLite Databases, Creating Content Providers, Using Content Providers, Adding Search to Your Application

Text Books:

1. J2ME: The Complete Reference, James Keogh, Tata McGrawHill, 2017.
2. Professional Android Application Development, Reto Meier, Wiley India, 2012.

Suggested Reading:

1. Mobile Design and Development, Brian Fling, O'Reilly, SPD, 2011.
2. Beginning Android Application Development, Wei-Meng Lee, Wiley Publishing, Inc, 2012
3. Android a Programming Guide, Jerome(J.F.) DiMarzio, McGrawHill, 2010
4. https://onlinecourses.nptel.ac.in/noc16_cs13
5. https://developer.android.com/index.html

16CSE05

COMPUTER GRAPHICS

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To Identify and explain the core concepts.
2. To Acquire knowledge about device level algorithms for displaying two dimensional output primitives for raster graphics system.
3. To Acquire knowledge about the basic concepts of representing 3D objects in 2D.
4. To Introduce computer graphics techniques transformations, clipping, curves and surfaces.

Course Outcomes:

1. Review the core concepts of computer graphics.
2. Analyse graphics techniques for rasterization, clipping, curve generation etc.
3. Evaluate pictures using various algorithms.
4. Understand the pipeline of typical graphics
5. Interpret and apply relevant problem solving methodologies

Prerequisites:
Knowledge of Linear Algebra (vectors and matrices), Good programming skills

UNIT-I

Graphics Systems and Models: Graphics system, Images, Physical and synthetic, Imaging system, synthetic camera model, programming interface, graphics architectures programmable pipelines, performance characteristics.

Graphics Programming: Programming two-dimensional applications, OpenGL API, Primitives and attributes, color, viewing, control functions

UNIT-II

Input and Interaction: Input device, clients and servers, displays lists, display lists and modeling, programming event driven input, picking, building interactive models, animating Interactive programs, logic operations.

Geometrics Objects: Three - dimensional primitives, coordinates systems and frames, frames in OpenGL, Modeling colored cube.

UNIT-III

Transformations: Affine Transformations, Transformations in homogenous coordinates, concatenation of Transformations, OpenGL transformation matrices, **Viewing:** Classical and Computer views, Viewing with a computer, Positioning of camera, Simple projections, Projections in OpenGL, Hidden surface removal, Parallel-projection matrices, Perspective projection matrices.

UNIT-IV

Lighting and Shading: Light sources, The Phong lighting model, Computational vectors, Polygonal shading, Light sources in OpenGL, Specification of matrices in OpenGL, Global illumination.

From Vertices To Frames: Basic implementation strategies, line-segment clipping, polygon clipping, clipping of other primitives, clipping in three dimensions, Rasterization, Bresenham's algorithm, Polygon Rasterization, Hidden surface removal, anti-aliasing, display considerations.

UNIT-V

Modelling & Hierarchy: Hierarchal models, trees and traversal, use of tree data structure; animation, Graphical objects, Scene graphs, Simple scene graph API, Open Scene graph, other tree structures.

Curves and Surfaces: Representation of curves and surfaces, design criteria, Bezier curves and surfaces, Cubic B-splines, General B-splines, rendering curves and surfaces, curves and surfaces in OpenGL.

Text Books:

1. Edward Angel, Computer Graphics A Top-Down Approach with shader based openGL, Peasrson Education, 6th edition -2011.
2. Hearn Donald, Pauline Baker M: Computer Graphics with openGL, 4th Edition, Prentice Hall PTR, 2010.
3. Fransis S Hill Jr., Stephen M Kelley, Computer Graphics Using OpenGL, Prentice-Hall Inc., 3rd Edition , 2007.

Suggested Reading &Online Resources:

1. Edward Angel ,Computer Graphics A Top-Down Approach using openGL, Peasrson Education, 5th edition -2009.
2. Jim X. Chen, Foundation of 3D Graphics Programming Using JOGL and Java3D, Springe Verlag, 2006.
3. Hearn Donald, Pauline Baker M: Computer Graphics, 2nd edition , Prentice Hall PTR, 1995.
4. http://nptel.ac.in/courses/106106090/
5. http://nptel.ac.in/courses/106102065/

16CSE06

ADVANCED COMPUTER ARCHITECTURE

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course objectives:

1. To provide concepts on performance measurement of processor architectures
2. To provide knowledge about the need of parallel processing
3. To provide basics about parallelism techniques implemented in uniprocessor technologies.
4. To gain knowledge of state-of-the art technologies like superscalar and vector processor
5. To gain knowledge on multiprocessor and multi-core technologies.

Course Outcomes:

1. Acquire skills to measure the performance of various processor architectures
2. Apply parallel processing techniques
3. Gain knowledge on parallelism techniques implemented in uniprocessor technologies.
4. Understand the state-of-the art technologies like superscalar and vector processor
5. Gain knowledge multiprocessor and multi-core technologies.
6. Understand the parallel program development.

UNIT-I

Measuring Performance and cost: Performance measurement, Enhancements to Uniprocesssor models, Benchmarks, Basic model of advanced computer architectures.

UNIT-II

Pipelining and superscalar techniques: Basic pipelining, data and control hazards, Dynamic instruction scheduling, Branch prediction techniques, Performance evaluation, Case study- Sun Microsystems - Microprocessor.

UNIT-III

Vector Processors: Vector Processor Models, Vector architecture and Design, Performance evaluation, and Programming Vector processors.

Array Processors: Parallel array processor model, and Memory organization. Interconnection networks: performance measures, static and dynamic topologies.

UNIT-IV

Multiprocessors and Multi computers: Multiprocessor models, Shared-memory and Distributed memory architectures, Memory organization, Cache Coherence and Synchronization Mechanisms, Parallel computer, and Performance models.

UNIT-V

Software for parallel Programming: Parallel models, languages, and compilers, Parallel Program Development and Environments, and Trends in Parallel systems-Heterogeneous Computing multi-core architectures, and Asymmetric multi-core architectures.

Text Books:

1. John L. Hennessey and David A. Patterson , "Computer Architecture, A QuantitativeApproach ", 4 th Edition, Elsevier, 2007.

Suggested Reading:

1. Sajjan G. Shiva, "Advance Computer Architecture ", Taylor Series Group, CRC press, 2006.
2. Kai Hwang and Naresh Jotwani, "Advanced Computer Architecture", Mc Graw Hill, 1999.

16CSC22

OPERATING SYSTEMS LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. To understand the design aspects of operating system.
2. To design and apply the process management concepts.
3. To design and apply the storage management concepts.

Course Outcome:

1. To use Unix utilities and perform basic shell control of the utilities.
2. To use the Unix file system and file access control.
3. To write programs systems based on multiple cooperating processes or threads.
4. To implement process scheduling, synchronization and memory management algorithms.
5. To implement process synchronization problems.
6. To implement process deadlocks.

List of experiments:

1. Programs using LINUX shell scripts.
2. Programs using process related system calls.
3. Programs to illustrate threads.
4. Implement CPU scheduling algorithms (a) Round Robin (b) SJF (c) FCFS.
5. Echo server using pipes.
6. Echo server using messages.
7. Producer- Consumer problem using shared memory.
8. Dining philosopher problem using semaphore.
9. Implement page replacement algorithms (a) FIFO (b) LRU.
10. Bankers algorithm for Deadlock detection and avoidance.
11. Programs to illustrate different file related System calls.
12. Printing file flags for specified descriptor.

Text Books:

1. Deitel and Deitel, “Operating System”, Pearson Education, New Delhi, Third Edition, 2007.
2. K A Robbin and Steve Robbins “UNIX Systems Programing”, PHI, 2003

16CSC23

DATA COMMUNICATION AND COMPUTER NETWORKS LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. Understand different types of network medium and devices.
2. Learn basic network commands.
3. Installation and working of simulation tools.
4. Performance measurement of network.
5. Create network topologies using simulation tools.

Course Outcomes:

1. Become familiar with different types of equipment and cables used in the networks lab.
2. Identification of various network devices.
3. Familiarity of basic network commands.
4. Ability to assign an IP address to a PC.
5. Ability to connect a PC to the LAN.
6. Design network topologies using simulation tools.

List of Experiments:

1. Study of Network medium and devices.
2. Study of basic network commands and configuration tools (*ifconfig, ping, traceroute, nslookup,dig,arp,netstat, nmap* etc.,).
3. Introduction to Network Simulation tools and Installation of any one tool
4. Simulation of a simple network with three nodes and identifying as a central node.
5. Study and simulation of any two topologies.
6. Simulation of a network with multiple routers and nodes by using hybrid topology.
7. Installation and configuration of NetAnim.
8. Implementation of FTP using TCP bulk transfer.
9. Calculation of the performance for the network implemented in experiment.
10. Analysis of network traces using Wireshark or any tool.

Text Booksan Suggested Reading:

1. https://www.nsnam.org/docs/release/3.18/tutorial/ns-3-tutorial.pdf
2. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fifth Edition, 2013

16CSC24

SOFTWARE ENGINEERING LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. To identify Project Scope, Objectives and infrastructure.
2. To understand Software Engineering methodologies for project development.
3. To gain knowledge about Computer Aided Software Engineering (CASE) tools.
4. To use effective communication skills and technical skills to assure production of quality software.

Course Outcomes:

1. To identify the problem scope and constraints of the problem.
2. To analyze the requirements for the system to be developed.
3. To excel in designing the system.
4. To work with CASE tools.
5. To gain knowledge in verifying and validating the product.
6. To develop a mini project.

A group of five students are identified as a team and the team should be able to develop mini project on the case studies like:

(i) Online Library Management system in college.

(ii) Online Feedback system in college.

(iii) Online Leave Management System for staff in college

(iv) Online Attendance Management system in college.

(v) Online Canteen Management System in college.

The team need to do the following experiments to develop the mini project.

Week 1: Introduction to Software product Development and Tools.
Week 2: Problem Definition.
Week 3: Software Requirement Specification-Standard IEEE SRS document.
Week 4: Data dictionary.
Week 5,6: System Design-structural diagrams, UML diagrams.
Week 7,8: Implementation using Computer Aided Software Engineering tools(CASE).

Week 9: Generating Test Cases.
Week 10: Document Writing.

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education, 2007
2. Roger S. Pressman, "Software Engineering –A Practitioners Approach", 7th Edition, Pearson Education, India, 2010

Suggested Reading:

1. ftp://public.dhe.ibm.com/software/rational/web/datasheets/rose_ds.pdf
2. http://docs.staruml.io/en/latest/

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
SCHEME OF INSTRUCTION AND EXAMINATION
V-Semester of B.E under CBCS
COMPUTER SCIENCE AND ENGINEERING

SEMESTER-VI

Sl.No	Syllabus Ref. No	SUBJECT	Scheme of Instruction		Scheme of Examination			Credits
			Periods per Week		Duration Credits of SEE in Hours	Maximum Marks		
			L/T	P/D		CIE	SEE	
THEORY								
1	16CSC25	Compiler Construction	3	-	3	30	70	3
2	16CSC26	Artificial Intelligence	3	-	3	30	70	3
3	16CSC27	Mobile Computing	3	-	3	30	70	3
4	16CSC28	Information and Network Security	3	-	3	30	70	3
5	16CSC29	Internet of Things	3	-	3	30	70	3
6	16CSE 07/08/09	Elective-III	3	-	3	30	70	3
PRACTICALS								
7	16CSC30	Information and Network security Lab	-	3	3	25	50	2
8	16CSC31	Internet of Things Lab	-	3	3	25	50	2
9	16CSC32	Mini Project-II	-	3	3	50	-	1
		TOTAL	18	9	-	280	520	23

Elective-III:
16CSE07 – Computer Vision
16CSE08 – Soft Computing
16CSE09 – Data Mining

L: Lecture T: Tutorial D: Drawing P: Practical
CIE - Continuous Internal Evaluation SEE - Semester End Examination

16CSC25

COMPILER CONSTRUCTION

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To implement the concept learned in automata theory and languages to the field of Computer Science.
2. Analyze the basic steps involved in converting a source language to target code.
3. Understands the concepts of parsers and can write solutions for various grammars by using tools, and also analyzes different storage techniques, error recovery strategies.
4. Gain the knowledge to write a compiler program or can able to build a compiler.

Course Outcomes:

1. Identify the basic concepts needed for the development of a compiler
2. Analyze the various phases and Tools of a Compiler
3. Describe the differences between Top-Down and Bottom-Up Parsers and apply parsing methods for various grammars.
4. Compare and Contrast Symbol table organization for Block Structured and Non-Block Structured languages.
5. Analyze the concepts involved in Intermediate, Code Generation and Code Optimization Process.
6. Recognize the various types of errors and error recovery strategies in phases of Compilation.

UNIT-I

Introduction – Programs related to compilers. Translation process.Major data structures.Other issues in compiler structure. Boot strapping and porting.
Lexical analysis – The role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, The Lexical-Analyzer Generator Lex.

UNIT-II

Syntax Analysis – Introduction, Top-Down parsing, Brute Forcing, Recursive Descent, Predicative LL(1), Bottom-Up parsing, Introduction to LR Parsing, Powerful LR parsers SLR, CALR, LALR, Using Ambiguous Grammars,
Parser Generators - YACC.

UNIT-III

Syntax Directed Translation – Syntax Directed Definitions. Evaluation Orders for SDDs. Applications of Syntax Directed Translation.

Symbol Table Organization - Structure of Symbol table, Symbol Table organization for Block Structured and non-block Structure languages, Data Structures of symbol Table.

UNIT-IV

Intermediate code generation: Variants of syntax trees. Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow.

Storage Organization.Stack, Heap Management, Garbage Collection.

Code Generation – Issues in the Design of a Code Generator.The Target Language. Addresses in the Target Code Basic Blocks and Flow Graphs. Optimization of Basic Blocks.Peephole Optimization.

UNIT-V

Machine Independent Optimizations – The Principal Sources of Optimizations, Introduction to data flow analysis, Foundation of data flow analysis.

Error Recovery : Error detecting and Reporting in various Phases.

Introduction to Advanced Topics : Review of compiler structure, advanced issues in elementary topics, the importance of optimizations, Structure of optimizing compilers

Text Books:

1. Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D Ullman, "Compilers: Principles Techniques & Tools", Pearson Education 2nd Edition 2013.
2. Muchnik, "Advanced Compiler Design and Implementation", Kauffman(1998)

Suggested Reading:

1. Kenneth C Louden, "Compiler Construction: Principles and Practice", Cengage Learning. Lex &Yacc, John R Levine, Oreilly Publishers.
2. Keith D Cooper & Linda Tarezon, "Engineering a Compiler", Morgan Kafman, Second edition. Lex &Yacc, John R Levine, Tony Mason, Doug Brown, Shroff Publishers.

Online Resources:

1. http://www.nptel.ac.in/courses/106108052
2. https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/
3. http://en.wikibooks.org/wiki/Compiler_Construction
4. http://dinosaur.compilertools.net/
5. http://epaperpress.com/lexandyacc/

16CSC26

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To list the significance of AI.
2. To discuss the various components that are involved in solving an AI problem.
3. To analyze the various knowledge representation schemes, Reasoning and Learning techniques of AI.
4. Apply the AI concepts to build an expert system to solve the real world problems.

Course Outcomes:

1. Differentiate between a rudimentary Problem and an AI problem, it's Characteristics and problem solving Techniques.
2. Determine and evaluate the various search strategies.
3. Compare and contrast the various "knowledge representation" schemes of AI.
4. Understand and Analyze the various reasoning techniques involved in solving AI problems.
5. Understand the different learning techniques.
6. Apply the AI techniques to solve the real world problems using Prolog.

UNIT I

Intelligent Agents: Intelligent agents**,** structure of agents

Introduction & Problem Solving: AI problems, AI Technique, Defining problem as a State-Space Search, Production Systems, Problem Characteristics.

Heuristic Search Techniques: Generate–and–test, Hill Climbing, Best–First Search,Problem Reduction, Constraint Satisfaction.

UNIT II

Game Playing: Overview, Min-Max search Procedure, Adding Alpha-beta Cutoffs, AdditionalRefinements, Iterative Deepening.

Using Predicate Logic: Representing simple facts in logic, Representing Instance and ISARelationships, Computable Functions, propositional calculus and predicates, Resolution.

UNIT III

Uncertainty and Reasoning Techniques: Non monotonic reasoning, Logics for Non monotonicreasoning, Implementation issues.

Statistical reasoning: Probability and Bayes theorem, Certainty factors and Rule-based systems,Bayesian Networks, Dempster-Shafer Theory.

UNIT IV

Learning: Introduction, Rote learning, Learning by taking advice, learning in problem solving, learning from examples: Induction.

Expert System: Representing and Using Domain Knowledge, Expert systems shells, Explanation, Knowledge Acquisition.

UNIT V

Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Statistical NLP, Spell Checking.

PROLOG-The Natural Language of AI: Prolog facts and rules, variables, control structures, arithmetic operators, matching in Prolog, backtracking.

Text Books:

1. Elaine Rich, Kevin Night, Shivashankar B Nair, "Artificial Intelligence",3rd Edition.,2008
2. Russell Norvig, "Artificial Intelligence-Modern Approach", 3rd edition, 2009.

Suggested Reading:

1. Saroj Kaushik, "Artificial Intelligence", Cengage Learning India, 2012.
2. Nelson M. Mattos, "An Approach to Knowledge Base Management", Springer Berlin Heidelberg, 1991.

Online Resources:

1. http://nptel.ac.in/courses/106106126/
2. http://nptel.ac.in/courses/106105077/

MOBILE COMPUTING

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To impart fundamental concepts in the area of mobile computing.
2. To provide a computer systems perspective on the converging areas of wireless and mobile applications.
3. To study the specification and functionalities of various protocols / standards of mobile networks.
4. To understand transactions and mobility support.

Course Outcomes:

1. Gain knowledge in the fundamental concepts of mobile and wireless applications.
2. Understand the principles of wireless transmission and cellular networks.
3. Compare various telecommunication systems and broadcasting techniques.
4. Identify various wireless LAN and routing protocols for different environments.
5. Understand file systems and transaction for mobility support.
6. Will have an understanding of issues related to privacy and ethics.

UNIT-I

Introduction: Applications,Wireless Transmission, Frequencies for Radio Transmission, Signals, Antennas, Signal Propagation, Cellular Systems, Spread Spectrum.

MediumAccess Control: Motivation for Specialized MAC, SDMA, FDMA, TDMA, CDMA; Comparison of S/F/T and CDMA.

UNIT-II

Telecommunication Systems: GSM, GPRS.

Satellite Systems: Applications, Basics, Parameters and Configurations, Capacity Allocation, FAMA and DAMA, Broadcast Systems, DAB, DVB, CDMA,3G and4G.

UNIT-III

Wireless LAN: IEEE 802.11 Architecture, Services, MAC – Physical Layer, IEEE 802.11a – 802.11b standards, Bluetooth, HIPER LAN.

UNIT-IV

Mobile Adhoc Networks: Routing in Ad-hoc Networks, Routing Protocols, Destination Sequenced Distance Vector, Dynamic Source Routing, Least Interference Routing, Cluster Based Gateway Switch Routing, Global State Routing, Fish-eye state Routing, Location Aided Routing.

Mobile Network Layer: Mobile IP - Dynamic Host Configuration Protocol.

Mobile Transport Layer: Traditional TCP**,** Classical TCP Improvements.

UNIT-V

Publishing & Accessing Data in Air: Pull and Push Based Data Delivery models,Data Dissemination by Broadcast, Directory Service in Air, Energy Efficient Indexing scheme for Push Based Data Delivery.

Support for Mobility: File Sharing for Mobility support, Coda,WAP, Sync MC, WAP 2.0

Mobile Transaction: Models for Mobile Transaction, Kangaroo and Joey transactions, Team Transaction, Recovery Model for Mobile Transactions, Mobile Privacy and Ethics.

Text Books:

1. Jochen Schiller, Mobile Communications, Pearson Education, 2nd Edition, 2009.
2. Kurnkum Garg, Mobile Computing : Theory and Practice, Pearson Education , 2010
3. Raj Kamal,"Mobile Computing", Oxford University Press, 2nd edition,2014.

Suggested Reading:

1. S. Acharya, M. Franklin and S. Zdonil, "Balancing Push and Pull for Data Broadcast, Proceedings of the ACM SIGMOD", Tuscon, AZ, May 1997.
2. Rappaport, Wireless Communications: Principles and Practice, Pearson Education, 2nd Edition 2010.
3. Prasant Kumar Pattnaik, Rajib Mall, "Fundamentals of Mobile Computing", PHI, 2012
4. "A Survey of Mobile Transactions appeared in Distributed and Parallel databases" 16,193- 230, 2004, Kluwer Academics Publishers.
5. AsokeK Talukder, Roopa R Yavagal, Mobile Computing, TMH 2008.

INFORMATION AND NETWORK SECURITY

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. Deal with the underlying principles of information and network security.
2. To understand the network security, services, attacks, mechanisms, types of attacks on TCP/IP protocol suite.
3. To comprehend and apply network layer security protocols, Transport layer security protocols, Web security protocols.
4. To comprehend and apply authentication services, authentication algorithms.
5. Deal with the key exchange problem and solutions using the Diffie-Hellman and Message Authentication Codes (MAC) and signature schemes.

Course Outcomes:

1. Understand the most common type of information and network threat sources.
2. Be able to determine appropriate mechanisms for protecting the network.
3. Design a security solution for a given application system with respect to security of the system.
4. Understand the information and network security issues and apply the related concepts for protection and communication privacy.
5. Understand application security using smart- cards.
6. Understand the operation of e-payments, micro- payments and related security issues, protocols.

UNIT-I

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

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

UNIT-II

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

UNIT-III

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

UNIT-IV

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

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

UNIT-V

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

Text Books:

1. Michael E. Whitman and Herbert J. Mattord: Principles of Information Security, 6thEdition, Cengage Learning, 2017.
2. William Stallings: Cryptography and Network Security, 7thEdition, Pearson Education, 2015.

Suggested Reading:

1. Behrouz A. Forouzan “Cryptography and its principles”.McGraHill, 3rd Edition, 2016.

16CSC29

INTERNET OF THINGS

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. Understand vision and Introduction to IoT.
2. Explore Data and Knowledge Management and use of Devices in IoT Technology.
3. Understand State of the Art – IoT Architecture.
4. Understand IoT protocols.
5. Programming with Raspberry Pi.
6. Explore the Real World IoT Design Constraints, Industrial Automation and Commercial Building Automation in IoT.

Course Outcomes:

1. Understand the Architectural Overview of IoT
2. Use of Devices, Gateways and Data Management in IoT.
3. Building state of the art architecture in IoT.
4. Understand various protocols used in IoT.
5. Understand Application of IoT in Industrial and Commercial Building Automation.
6. Understand Real World Design Constraints.

UNIT-I

Overview: IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. M2M and IoT Technology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management.

UNIT-II

IoT Reference Architecture: IoT Architecture-State of the Art – Introduction, State of the art, Reference Model and architecture, IoT reference Model - IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints- Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control.

UNIT-III

IoT Protocols:Infrastructure (ex: 6LowPAN, IPv4/IPv6, RPL), **Identification** (ex: EPC, uCode, IPv6, URIs), **Comms / Transport** (ex: Wifi, Bluetooth, LPWAN), **Discovery** (ex: Physical Web, mDNS, DNS-SD), **Data Protocols** (ex: MQTT, CoAP, AMQP, Websocket, Node), **Device Management** (ex: TR-069, OMA-DM), **Semantic** (ex: JSON-LD, Web Thing Model), **Multi-layer Frameworks** (ex: Alljoyn, IoTivity, Weave, Homekit).

Unit-IV

Raspberry Pi: Exemplary Device: RaspberryPi, About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry Pi with Python. NODEMCU (ESP8266) : Introduction and Architecture.

Unit-V

Domain Specific IOTs: Introduction, Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry.

Text Books:

1. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, **"**From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence**",** 1st Edition, Academic Press, 2014.
2. ArshdeepBahga, Vijay Madisetti, Internet of Things: A hands on approach, 2014, VPT publishers.

Suggested Reading:

1. Peter Waher**,** "Learning Internet of Things", PACKT publishing, BIRMINGHAM – MUMBAI
2. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer
3. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications.

Online Resources:

1. https://www.postscapes.com/internet-of-things-protocols/
2. https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/

16CSE 07

COMPUTER VISION

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To develop algorithms and techniques to analyze and interpret the visible world around us.
2. To understand the Fundamental Concepts Related To Multi-Dimensional Signal Processing.
3. To understand Feature Extraction algorithms.
4. To analyze Patterns in images.
5. To understand Visual Geometric Modeling.
6. To understand Stochastic Optimization.

Course Outcomes:

1. To understand concepts necessary in this field, to explore and contribute to research and further developments in the field of computer vision.
2. To apply in the field of Biometrics, Medical diagnosis, document processing, mining of visual content, to surveillance, advanced rendering etc.

Unit-I

Introduction to Computer Vision and Image Formation: Introduction, Geometric primitives and transformations,Photometric image formation, Digital Camera image formation.

Image Processing: Point operators , Linear filtering , More neighborhood operators, Fourier transforms, Pyramids and wavelets, Geometric transformations, Global optimization.

Unit-II

Feature detection and matching: Points and patches, Edges ,Lines.

Segmentation: Active contours ,Split and merge ,Mean shift and mode finding, Normalized cuts , Graph cuts and energy-based methods.

Feature-based alignment: 2D and 3D feature-based alignment ,Pose estimation Geometric intrinsic calibration.

Unit-III

Structure from motion: Triangulation, Two-frame structure from motion, Factorization, Bundle adjustment, Constrained structure and motion.

Dense motion estimation: Translational alignment, Parametric motion, Spline-based motion, Optical flow, Layered motion.

Unit-IV

Recognition: Object detection, Face recognition, Instance recognition, Category recognition, Context and scene understanding, Recognition databases and test sets

Unit-V

3D reconstruction : Shape from X, Active range finding, Surface representations, Point-based representations, Volumetric representations, Model-based reconstruction, Recovering texture maps.

Image-based rendering : View interpolation, Layered depth images, Light fields and Lumigraphs, Environment mattes, Video-based rendering.

Text Books:

1. "Computer Vision: Algorithms and Applications" by Richard Szeliski; Springer-Verlag London Limited 2011 .
2. Digital Image Processing"; R. C. Gonzalez and R. E. Woods; Addison Wesley;2008.

References

1. "Pattern Recognition: Statistical, Structural and Neural Approaches"; Robert J. Schallkoff; John Wiley and Sons; 1992.
2. "Computer Vision: A Modern Approach"; D. A. Forsyth and J. Ponce; Pearson Education; 2003.
3. Multiple View geometry. R. Hartley and A. Zisserman. 2002 Cambridge university Press
4. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
5. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.

Online links

1. CV online: http://homepages.inf.ed.ac.uk/rbf/CVonline
2. Computer Vision Homepage: http://www2.cs.cmu.edu/afs/cs/project/cil/ftp/ html/vision.html.

16CSE 08

SOFT COMPUTING

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To learn various types of soft computing techniques and their applications.
2. To acquire the knowledge of neural network architectures, learning methods and algorithms.
3. To understand Fuzzy logic, Genetic algorithms and their applications.

Course Outcomes:

1. Understand various soft computing techniques.
2. Understand various learning models.
3. Design and develop various Neural Network Architectures.
4. Understand approximate reasoning using fuzzy logic.
5. Analyze and design Genetic algorithms in different applications.
6. Ability to apply soft computing techniques to solve different applications.

UNIT-I

Soft computing vs. Hard computing, Various types of soft computing techniques.
Artificial Neural Networks: Fundamental concepts, Evolution of neural networks, Basic models of artificial neural network, Important terminologies of ANNs. McCulloch-Pitts neuron, Linear separability, Hebb network.

UNIT-II

Supervised Learning Neural Networks: Perceptron networks, Adaptive linear neuron (Adaline), Multiple Adaptive linear neuron (Madaline), Back propagation network.

UNIT-III

Unsupervised Learning Neural Networks: Kohonen Self Organizing networks, Adaptive resonance theory.
Associate Memory Networks: Bidirectional associative memory network, Hopfield networks.

UNIT-IV

Fuzzy Logic: Introduction to classical sets and Fuzzy sets, Fuzzy relations, Tolerance and equivalence relations, Membership functions, Defuzzification.

UNIT-V

Genetic Algorithms: Introduction, Basic operators and terminology, Traditional algorithm vs. genetic algorithm, Simple genetic algorithm, General genetic algorithm, Classification of genetic algorithm, Genetic programming, Applications of genetic algorithm.

Text Books:

1. S.N. Sivanandam& S.N. Deepa, "Principles of soft computing", Wiley publications, 2nd Edition, 2011.

Suggested Readings:

1. S. Rajasekaran& G.A. Vijayalakshmipai, "Neural Networks, Fuzzy logic & Genetic Algorithms, Synthesis & Applications", PHI publication, 2008.
2. LiMin Fu, "Neural Networks in Computer Intelligence", McGraw-Hill edition, 1994.
3. K.L.Du& M.N.S Swamy, "Neural Networks in a Soft Computing Framework", Springer International edition, 2008.
4. Simon Haykins, "Neural Networks a Comprehensive Foundation", PHI, second edition.
5. Goldberg, David E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, New Delhi, 2002.
6. N.P. Padhy and S.P. Simon,"Soft Computing: With Matlab Programming", Oxford University Press, 2015.

Online Resources:

1. https://onlinecourses.nptel.ac.in/noc18_cs13/preview.

16CSE09

DATA MINING

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To understand the principles of Data Warehousing and Data Mining.
2. To be familiar with the Data Warehouse Architecture and its implementation.
3. Learn how to produce a quantitative analysis report/memo with the necessary information to make decisions.
4. Provide understanding of mathematical concepts and algorithms used in data mining.
5. Identifying business applications of data mining
6. Develop and apply critical thinking, problem-solving, and decision-making skills.

Course Outcomes:

1. Understand the process, issues and challenges of knowledge discovery
2. Identify and analyze the significance and working of various data preprocessing methods.
3. Learn the architecture of data warehouse and its implementation through multi-dimensional modeling.
4. Understand operational database, warehousing and multidimensional need of data base to meet industrial needs.
5. Explore the concepts of market basket analysis to generate association rules.
6. Analyze and Evaluate the performance of Classification and Clustering algorithms
7. Understand the significance and methodologies of outlier detection schemes.

UNIT-I

Introduction: Fundamentals of data mining, Data Mining Functionalities, Issues in Data Mining.

Data Objects and Attribute types, Basic Statistical descriptions of data, Data Visualization, Measuring data similarity and Dissimilarity.

Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

UNIT-II

Data Warehouse and Online Analytical Processing: Basic Concepts of Data Warehouse, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Architecture, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization by Attribute-Oriented Induction.

Data Cube Computation: Preliminary Concepts, Data Cube Computation Methods.

UNIT-III

Mining Frequent Patterns, Associations and Correlations: Basic Concepts and Methods, Frequent Item set Mining Methods, Pattern Evaluation Methods: From Association Analysis to Correlation Analysis.

UNIT-IV

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Model Evaluation and Selection, Techniques to improve Classification Accuracy, Classification by Back propagation Prediction, Support Vector Machines, Lazy Learners.

UNIT-V

Cluster Analysis: Basic Concepts and Methods, Partitioning Methods: K-means Technique, Hierarchical Methods: Agglomerative and Divisive, Density Based Methods: DBSCAN technique, Evaluation of Clustering.

Outlier Detection: Outlier Detection Methods, Statistical Approaches, Proximity-Based Approaches, Clustering Based Approaches.

Text Books:

1. Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining – Concepts and Techniques", 3rd edition, Morgan Kaufmann Publishers, ELSEVIER, 2013.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar "Introduction to Data Mining", Pearson Education, 2006.

Suggested Reading:

1. Sam Aanhory& Dennis Murray "Data Warehousing in the Real World", Pearson Edn Asia.
2. K.P.Soman, S.Diwakar, V.Ajay ,"Insight into Data Mining", PHI, 2008.
3. Ralph Kimball Wiley "The Data Warehouse Life cycle Tool kit", student edition
4. William H Inmon, John Wiley & Sons Inc "Building the Data Warehouse", 2005.
5. Margaret H Dunham "Data Mining Introductory and advanced topics", Pearson education.
6. Arun K Pujari "Data Mining Techniques", 2nd edition, Universities Press.

16CSC30

INFORMATION AND NETWORK SECURITY LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. Understand basic cryptography principles, including some well-known algorithms for symmetric and public key encryption, digital signatures, key management.
2. To provide a practical exposure of both the principles and practice of advanced cryptography.
3. Understand and fulfill the requirements C.I.A.
4. Understand the underlying principles of information and network security.

Course Outcomes:

1. Demonstrate detailed knowledge of the role of encryption to protect data.
2. Analyze security issues arising from the use of certain types of technologies.
3. Master protocols for security services.
4. Master on the key exchange and Authentication protocols.

List of Programs:

1. To perform encryption and decryption using the following algorithms
   a. Ceaser cipher b. Substitution cipher c. Hill Cipher
2. Implement the DES algorithm logic in C.
3. Implement the DES algorithm logic in JAVA.
4. JAVA program that contains functions, which accept a key and input text to be encrypted/decrypted. This program should use the key to encrypt/ decrypt the input by using the triple DES algorithm. Make use of Java Cryptography package.
5. Implement the Blowfish algorithm logic.
6. Implement RSA algorithm.
7. Implement Message Authentication Code (MAC).
8. Calculate the message digest of a text using the SHA-1 algorithm.
9. Calculate the message digest of a text using the MD5 algorithm.
10. Explore the Java classes related to digital certificates.

11. Create a digital certificate of your own by using any tool.
12. Create the awareness on open SSL.

Suggested Readings:

1. Michael Gregg "Build Your Own Security Lab", Wiley India.
2. Cryptography and Network Security Principles and Practice, William Stallings, 5th Edition, Prentice Hall,2011
3. Alfred Basta, Wolf Halton, "Computer Security, concepts, issues and implementation: Cengage Learning".

16CSC31

INTERNET OF THINGS LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. To understand how sensors are used in IoT systems.
2. To understand how to program on embedded and mobile platforms including ESP8266 and Raspberry-Pi.
3. To understand how to communicate with mobile devices using various communication platforms such as Bluetooth and Wi-Fi.
4. To understand how to make sensor data available on the Internet.
5. To understand how to analyze and visualize sensor data.
6. To understand how to work as a team and create end-to-end IoT applications.

Course Outcomes:

1. Use different types of sensors in various IoT Systems.
2. Use of devices, Gateways and Database Management in IoT.
3. Working with ESP8266 to implement various IoT systems.
4. Working with Raspberry-Pi to implement various IoT Systems.
5. Understand Application of IoT in Agriculture and Industries.
6. Understand Real World Design Constraints.

List of Experiments:

1. Implementation of Home Automation System using WiFi Module.
2. Design and develop Rain Sensing Automatic Wiper System.
3. Develop a system to identify accident and send alert messages.
4. Implementation of Traffic Light System based on density, to decrease congestion.
5. Design and develop IoT Solar Power Monitoring System.
6. Design and develop patient health monitoring system.
7. Design and develop IoT based Fire Altering System to give alert message to fire department.
8. Implementation of Smart Agriculture Monitoring System.

Suggesting Reading :

1. ArshdeepBahga, Vijay Madisetti, Internet of Things: A hands on approach, 2014, VPT publishers;

Reference Books:

1. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications.

16CSC32

MINI PROJECT-II

Instruction	3 Hours per week
Duration of Semester End Examination	-
Semester End Examination	-
CIE	50 Marks
Credits	1

The students are required to carry out mini projects in any of the areas such as Design and Analysis of Algorithms, Automata Languages and Computation, Operating Systems, Data Communication and Computer Networks, Software Engineering, Compiler construction, Artificial Intelligence and Mobile Computing etc.

Students are required to submit a report on the mini project at the end of the semester.