DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Scheme of Instruction

and

Syllabi of

M.TECH (EMBEDDED SYSTEMS AND COMPUTING)

2019-2020

UNIVERSITY COLLEGE OF ENGINEERING

(AUTONOMOUS)

OSMANIA UNIVERSITY

HYDERABAD – 500 007, TELANGANA

SCHEME OF INSTRUCTION
M.TECH (EMBEDDED SYSTEMS AND COMPUTING)
Proposed from the Academic year 2019-20
SEMESTER - I

S.No	Course Code	Course Title	Scheme of Instruction		Contact Hrs/Wk	Scheme of Examination		Credits
			L/T	P		CIE	SEE	
1.	CS 101	Program Core I- Mathematical foundations of Computer Science	3	--	3	30	70	3
2.	CS 102	Program Core II- Advanced Data Structures	3	--	3	30	70	3
3.	CS 301	Program Core III- Embedded System Design	3	--	3	30	70	3
4.	CS 214	Program Elective I- Scripting Languages for design automation	3	--	3	30	70	3
5.	CS 100	Research Methodology in Computer Science	3	--	3	30	70	3
6.	AC 101	Audit Course I	2	--	2	30	70	0
7.	CS 151	Laboratory – I Advanced Data Structures Lab	--	3	3	50	-	1.5
8.	CS 251	Laboratory – II Scripting Languages for Design Automation Lab	--	3	3	50	-	1.5
Total			**17**	6	**23**	**280**	**420**	**18**

SEMESTER - II

S.No	Course Code	Course Title	Scheme of Instruction		Contact Hrs/Wk	Scheme of Examination		Credits
			L/T	P		CIE	SEE	
1.	CS 302	Program Core IV- Hardware and Software Co-design	3	--	3	30	70	3
2.	Elective II	Elective II	3	--	3	30	70	3
3.	Elective III	Elective III	3	--	3	30	70	3
4.	Elective IV	Elective IV	3	--	3	30	70	3
5.	AC 107	Audit Course II	2	--	2	30	70	0
6.	CS 371	Mini Project with Seminar		6	6	50*	-	3
7.	CS 351	Laboratory III Hardware and Software Co-design	--	3	3	50	-	1.5
8.	Laboratory IV	Laboratory IV	--	3	3	50	-	1.5
Total			**14**	**12**	**26**	**300**	**350**	**18**

***Mini Project with Seminar Evaluation:** 25 marks to be awarded by Supervisor and 25 marks to be awarded by Viva-Voce committee comprising Head, Supervisor and an Examiner.*

SCHEME OF INSTRUCTION
M.TECH (EMBEDDED SYSTEMS AND COMPUTING)
Proposed from the Academic year 2018-19

SEMESTER III

S.No	Course Code	Course Title	Scheme of Instruction		Contact Hrs/Wk	Scheme of Examination		Credits
			L/T	P		CIE	SEE	
1.	Elective V	Elective V	3	-	3	30	70	3
2.	Open Elective	Open Elective	3	-	3	30	70	3
3.	CS 381	Major Project Phase I	--	20	20	100**		10
		Total	**6**	**20**	**26**	**160**	**140**	**16**

** ***Major Project Phase I Evaluation:*** *50 marks to be awarded by Supervisor and 50 marks to be awarded by Viva-Voce committee comprising Head, Supervisor and an Examiner.*

SEMESTER – IV

S.No	Course Code	Course Title	Scheme of Instruction		Contact Hrs/Wk	Scheme of Examination		Credits
			L/T	P		CIE	SEE	
1.	CS 382	Major Project Phase II	--	32	32	---	200	16
		Total	**--**	**32**	**32**	**---**	**200**	**16**

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

Audit course 1 & 2

AC 101 : English for Research Paper Writing
AC 102 : Disaster Management
AC 103 : Sanskrit for Technical Knowledge
AC 104: Value Education
AC 105: Constitution of India
AC 106 : Pedagogy Studies
AC 107 : Stress Management by Yoga
AC 108: Personality Development through Life Enlightenment Skills.

Open Elective

CS 901 : Business Analytics

CS 902 : Industrial Safety

CS 903 : Operations Research

CS 904 : Cost Management of Engineering Projects

CS 905 : Composite Materials

CS 906 : Waste to Energy

List of Core Subjects:

S.No	Course Code	Course Title
1	CS 101	Mathematical Foundation of Computer Science
2	CS 102	Advanced Data Structures
3	CS 301	Embedded System Design
4	CS 302	Hardware and Software Co-design

Mandatory Course :

S.No	Course Code	Course Title
1	CS 100	Research Methodology in Computer Science

List of Labs:

S.No	Course Code	Course Title
1	CS 151	Advanced Data Structures Lab
2	CS 251	Scripting Languages for Design Automation Lab
3	CS 351	Hardware and Software Co-design Lab
4	Laboratory IV	Laboratory IV

List of Elective Subjects:

S.No	Course Code	Course Title
1	CS 311	Digital System Design
2	CS 312	Microcontrollers for Embedded Systems
3	CS 313	Advanced Computer Architecture
4	CS 314	Embedded Programming
5	CS 315	Field Programmable Gate Arrays
6	CS 316	System On Chip Architecture
7	CS 317	Optimization Techniques
8	CS 318	Product Design and Quality Management
9	CS 319	Design for Testability
10	CS 320	DSP Architecture
11	CS 321	Graph Theory and its Applications
12	CS 111	Mobile Computing
13	CS 112	Real Time Systems
14	CS 114	Multimedia Technologies
15	CS 115	Data Mining
16	CS 116	Network Security
17	CS 117	Machine Learning
18	CS 121	Cloud Computing
19	CS 122	Soft Computing
20	CS 123	Artificial Neural Networks
21	CS 124	Software Project Management
22	CS 127	Reliability and Fault Tolerance
23	CS 131	Software Engineering for RTS
24	CS 132	Simulation and Modelling
25	CS 133	Advanced Operating Systems

26	CS 134	Object Oriented Software Engineering
27	CS 135	Distributed Computing
28	CS 136	Advanced Databases
29	CS 211	Parallel Algorithms
30	CS 212	Grid Computing
31	CS 213	Real Time Operating Systems
32	CS 214	Scripting Languages For Design Automation
33	CS 201	Parallel Computer Architecture
34	CS 202	Parallel Programming
35	CS 103	Advance Algorithms
36	CS 104	Artificial Intelligence

CS 101 Mathematical foundations of Computer Science

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

COURSE OBJECTIVE:

- To understand the mathematical fundamentals that is prerequisites for a variety of courses like Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems, Bioinformatics, Machine learning.
- To develop the understanding of the mathematical and logical basis to many modern techniques in information technology like machine learning, programming language design, and concurrency.
- To study various sampling and classification problems.

COURSE OUTCOMES :

At the end of the Course, Student would be :

- Able to apply the understanding of probability and distribution functions to solve various applications of Computer science .
- Able to solve sampling and classification problems
- Able to Infer and apply the various statistical models with suitable assessment based on various samples relevant in Computer Science
- Able to use Concepts of Graph theory and Solve combinatorial enumeration problems
- Able to create solutions by applying the mathematical techniques for solving engineering applications in computer science

Unit 1

Probability mass, density, and cumulative distribution functions, Parametric families of distributions, Expected value, variance, conditional expectation, Applications of the univariate and multivariate Central Limit Theorem, Probabilistic inequalities, Markov chains.

Unit 2

Random samples, sampling distributions of estimators**,** Methods of Moments and Maximum Likelihood**.**

Unit 3

Statistical inference**,** Introduction to multivariate statistical models: regression and classification problems, principal components analysis**,** The problem of over fitting model assessment.

Unit 4

Graph Theory: Isomorphism, Planar graphs, graph coloring, Hamilton circuits and Euler cycles. Permutations and Combinations with and without repetition. Specialized techniques to solve combinatorial enumeration problems.

Unit 5

Computer science and engineering applications

Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems, Bioinformatics, Machine learning.

Recent trends in various distribution functions in mathematical field of computer science for varying fields like bioinformatics, soft computing, and computer vision.

References

1. John Vince, Foundation Mathematics for Computer Science, Springer.

2. K. Trivedi.Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley.

3. M. Mitzenmacher and E. Upfal.Probability and Computing: Randomized Algorithms and Probabilistic Analysis.

4. Alan Tucker, Applied Combinatorics, Wiley

CS 102 Advanced Data Structures

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

COURSE OBJECTIVE:

- The student should be able to choose appropriate data structures, understand the ADT/libraries, and use it to design algorithms for a specific problem.
- Students should be able to understand the necessary mathematical abstraction to solve problems.
- To familiarize students with advanced paradigms and data structure used to solve algorithmic problems.
- Student should be able to come up with analysis of efficiency and proofs of correctness.

COURSE OUTCOMES :

After completion of course, students would be able to:

- Understand the implementation of symbol table using hashing techniques.
- Develop and analyze algorithms for red-black trees, B-trees and Splay trees.
- Develop algorithms for text processing applications.
- Identify suitable data structures and develop algorithms for computational geometry problems.

Unit 1

Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries.

Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.

Unit 2

Skip Lists: Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists

Unit 3

Trees: Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees

Unit 4

Text Processing: Sting Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.

Unit 5

Computational Geometry: One Dimensional Range Searching, Two Dimensional Range Searching, Constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quad trees, k-D Trees.

Recent Trends in Hashing, Trees, and various computational geometry methods for efficiently solving the new evolving problem

References:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004.

2. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.

CS 301 EMBEDDED SYSTEM DESIGN

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

UNIT-I

Introduction to Embedded Systems: Characteristics and quality attributes of Embedded Systems Challenges in Embedded System Design, Application and Domain specific Embedded Systems.

UNIT –II

Embedded System Architecture: Instruction Set Architecture, CISC and RISC instruction set architecture, Basic Embedded Processor/Microcontroller Architecture, C1SC Examples- Motorola (68HCI1), RISC Example- ARM, DSP Processors, Harvard Architecture Microcontroller Example - PIC.

UNIT -III

Embedded Hardware Design and Development: VLSI and Integrated Circuit Design, EDA tools, usage of EDA tools and PCB layout.

Embedded firmware and Design and Development: Embedded Firmware Design Approaches and Development languages and Programming in Embedded in C.

UNIT -IV

Introduction to Real Time Operating System: Tasks and Task States, Tasks and Data, Semaphores, and Shared Data; Message Queues, Mailboxes and Pipes, Timer functions, Events, Memory Management, Interrupt Routines in an RTOS Environment, OS Security Issues and Mobile OS.

UNIT-V

Embedded Systems Development Environment: IDE, Cross Compilation, Disassembler, Simulators, Emulators and Debugging, Target Hardware Debugging, Boundary Scan. Product Enclosure Design and Development Tools, Embedded Product Development Life Cycle- Different phases and Approaches' of EDLC. Trends in Embedded Industry.

Suggested Reading:

1. Shibu K V, *Introduction to Embedded Systems*, Tata McGraw HilI,2010.
2. Raj Kamal, *Embedded Systems Architecture, Programming & Design*, Tata McGraw Hill, 2010.
3. Dr K.V.K.K. Prasad, *Embedded/Real Time Systems: Concepts, Design and Programming*, Dreamtech Press, 2004.

CS 302 HARDWARE AND SOFTWARE CO-DESIGN

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

UNIT-I

Introduction: Issues in co-designs, models, architectures, languages and generic co-design methodology.

UNIT-II

Hardware / Software co-synthesis Algorithms: Introduction, architecture models, hardware/software partitioning, distributed system co-synthesis.

UNIT-III

Prototyping and Emulation: Introduction, prototyping and emulation techniques, prototyping and emulation environments and future developments in emulation and prototyping.

UNIT-IV

Compilation Techniques and Tools for Embedded processor architectures: Introduction, Modern embedded architectures, embedded software development needs, compilation techniques, practical considerations in a compiler development environment.

UNIT-V

Design Specification and Verification: Introduction, Concurrency, Coordinating Concurrent Computations, Interfacing Components and Verifications.

Suggested Reading:

1. Jargen Staunstrup, Wayne Wolf, *Hardware/Software Co-Design, Principles and Practice,* Kluwer Academic Publishers, 1997.
2. Schaumont, Patric R, *A Practical Introduction to Hardware/Software Codesign*, 2nd Edition, Springer publishers, 2013.

CS 100 Research Methodology In Computer Science

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Objectives:
The main aim is to enable the students
1. To understand the research process
2. To solve unfamiliar problems using scientific procedures
3. To pursue ethical research
4. To use appropriate tools for documentation and analysis of data

Course Outcomes:
At the end of this course, students will be able to
- understand research problem formulation
- design experiments
- analyze research related information
- write papers and thesis
- follow research ethics
- use tools for analysis and thesis writing

UNIT-I

Research Process: Meaning of Research, Objectives and Motivation of Research, Technological Innovation, Types of Research, Research Vs Scientific method, Research Methodology vs Research Methods, Research process.

Research Problem Formulation: Problem solving in Engineering, Identification of Research Topic, Problem Definition, Literature Survey, Literature Review.

Research Design: Research Design: What it is?, Why we need Research Design? Terminology and Basic Concepts, Different Research Designs, Experimental Designs, Important Experimental Designs, Design of Experimental Setup, Use of Standards and Codes.

UNIT-II

Mathematical Modeling: Models in General, Mathematical Model, Model Classification, Modeling of Engineering Systems.

Probability and Distributions: Importance of Statistics to Researchers, Probability Concepts, Probability Distributions, Popular Probability Distributions, Sampling Distributions.

Sample Design And Sampling: Sample design, Types of sample designs, The Standard Error, Sample Size for Experiments, Prior Determination Approach, Use of Automatic Stopping Rule.

Hypothesis Testing And ANOVA: Formulation of Hypothesis, Testing of Hypothesis, Analysis of Variance.

UNIT-III

Design of Experiments and Regression Analysis: Design of Experiments, Planning of Experiments, Multivariate Analysis, Simple Regression and Correlation, Multiple Regression and Correlation

Analysis and Interpretation of Data: Introduction, Data Checking, Data Analysis, Interpretation of Results, Guidelines in Interpretations.

Accuracy, Precision and Error Analysis: Introduction, Repeatability and Reproducibility, Error Definition and Classification, Analysis of Errors, Statistical Analysis of Errors, Identification of Limitations

UNIT-IV

Writing of Papers and Synopsis: Introduction, Audience Analysis,, Preparing Papers for Journals, Preparation of Synopsis of Research Work

Thesis Writing Mechanics: Introduction, Audience for Thesis Report, Steps in Writing the report, Mechanics of Writing, Presentation of graphs, figures and tables.

Structure of Thesis Report: Suggested Framework of the Report, Preliminary Pages, Main Body of Thesis, Summary, Appendices, References, Glossary.

UNIT-V :

Ethics in Research: Importance of Ethics in Research, Integrity in Research, Scientific Misconduct and Consequences.

Spreadsheet tool: Introduction, Quantitative Data Analysis Tools, Entering and preparing your data, Using statistical functions, Loading and using Data Analysis Tool Pack *[Tools: Microsoft Excel / Open office]*

Thesis writing & Scientific editing tool. [Tool: Latex]: Introduction, Document Structure, Typesetting Text, Tables, Figures, Equations, Inserting References

Suggested Reading:

1. R.Ganesan; Research Methodology for Engineers; MJP Publishers; Chennai, 2011.
2. Paul R Cohen. Empirical Methods in AI. PHI, New Delhi, 2004
3. C.R.Kothari, Research Methodology, Methods & Technique; New age International Publishers, 2004
4. Kumar, Ranjit. Research Methodology-A Step-by-Step Guide for Beginners, (2nd.ed), Singapore, Pearson Education, 2005
5. https://arxiv.org/pdf/physics/0601009.pdf
6. https://pdfs.semanticscholar.org/e1fa/ec8846289113fdeb840ff3f32d102e46fbff.pdf
7. LaTEX for Beginners, Workbook, Edition 5, March 2014.
8. Chapter 13, An introduction to using Microsoft Excel for quantitative data analysis: Management Research: Applying the Principles © 2015 Susan Rose, Nigel Spinks & Ana Isabel Canhoto.

CS 151 Advanced Data Structures lab

Credits: 1.5

Instruction: 3 hrs per week *CIE: 50 marks*

1. Write a program that implements stack and Queue operations using
    a. Arrays
    b. linked list
2. Write a program to perform the following operations on singly linked list and doubly linked list
    a. Creation
    b. Insertion
    c. Deletion
    d. Traversal.
3. Implement recursive and non recursive i) Linear search ii) Binary search
4. Study and Implementation of Different sorting algorithms and Find Time and Space complexities.
5. Implement Recursive functions to traverse the given binary tree in
    a. Preorder
    b. Inorder
    c. Postorder
6. Study and Implementation of different operations on
    a. Binary Search Tree
    b. AVL tree
    c. Red Black Tree
7. perform the following operations
    a. Insertion into a B-tree
    b. Deletion from a B-tree
8. Implement Different Collision Resolution Techniques.
9. Study and Implementation of Following String Matching algorithms:
    a. Rabin-Karp algorithm
    b. Knuth-Morris-Pratt algorithm
    c. Boyer-Moore algorithm

10. Implement the following using java:

1. Single Source Shortest Path algorithms
2. All pairs shortest path algorithms
3. Minimal Spanning Tree algorithms
4. String and Pattern matching algorithms
5. Maximum Flow/ Minimum cut algorithms

Note : The students have to submit a report at the end of the semester.

CS 251 Scripting Languages for Design Automation Lab

Instruction: 3 hrs per week *Credits: 1.5* *CIE: 50 marks*

S. No.	Program
1	Introduction to Python Programming: A. Running instructions in Interactive interpreter and a Python Script. B. Write a program to purposefully raise Indentation Error and Correct it C. Write a program to compute distance between two points taking input from the user D. Write a program add.py that takes 2 numbers as command line arguments and prints its sum. E. Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects F. Write a Program for checking whether the given number is a even number or not.
2	Control Structures, Lists A. Program to find the largest three integers using if-else B. Program that receives a series of positive numbers and display the numbers in order and their sum C. Program to find the product of two matrices $[A]_{mxp}$ and $[B]_{pxr}$ D. Program to display two random numbers that are to be added, the program should allow the student to enter the answer. E. If the answer is correct, a message of congratulations should be displayed. F. If the answer is incorrect, the correct answer should be displayed. G. Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . 1/10. H. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
3	Functions and Recursion A. Write recursive and non-recursive functions for the following B. To find GCD of two integers C. To find the factorial of positive integer D. To print Fibonacci Sequence up to given number n E. To display prime number from 2 to n. F. Function that accepts two arguments: a list and a number n. It displays all of the numbers in the list that are greater than n G. Functions that accept a string as an argument and return the number of vowels and consonants that the string contains
4	Files, Exceptions, Lists, Sets, Random Numbers A. Program to write a series of random numbers in a file from 1 to n and display. B. Program to write the content in a file and display it with a line number followed by a colon C. Program to display a list of all unique words in a text file D. Program to analyse the two text files using set operations E. Write a program to print each line of a file in reverse order. F. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file? G. Write a program combine lists that combines these lists into a dictionary.
5	Object Oriented Programming A. Program to implement the inheritance B. Program to implement the polymorphism
6	GUI Programming A. Program that converts temperature from Celsius to Fahrenheit B. Program that displays your details when a button is clicked C. Write a GUI for an Expression Calculator using tk

CS 311

Digital System Design

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Analysis & Design of Combinational Logic: Introduction to combinational circuits, code conversions, decoder, encoder, priority encoder, multiplexers as function generators, binary adder, subtractor, BCD adder, Binary comparator, arithmetic logic units.

UNIT-II

Design of Sequential Circuits- Derivation of State diagrams and tables, transition table, excitation table and equations. Analysis of simple synchronous sequential circuits, construction of state diagram, counter design with state equations, Registers , serial in serial out shift registers, tristate register, timing considerations.

UNIT-III

Minimization and Transformation of Sequential Machines: Simplification of incompletely specified machines-Flow table, State reduction, Merger graphs, Merger Tables, Compatible pairs, Minimal closed covers – Races, Cycles and Hazards.

UNIT-IV

Design options of Digital Systems: Programmable logic devices, programmable read only memory, programmable logic arrays and programmable array logic, Design using PLA, PAL, and Field Programmable Gate Arrays. **Synthesis**: Design flow of ASICs and FPGA based system.

UNIT-V

Introduction to Verilog HDLs –Modeling levels- Data types- Modules and ports- Instances – Basic Language concepts- Data flow modeling. **Design Examples**: Adders and Subtractors. Behavioral modeling, Gate-level modeling. Tasks and functions – Modeling techniques – logic synthesis with Verilog. **Design Examples**: Multiplication and Division Algorithms.

Suggested Reading:

1. Ming-Bo Lin., *Digital System Designs and Practices using Verilog HDL and FPGAs*, Wiley, 2nd Edition, 2015.
2. Michael D. Ciletti, *Advanced Digital Design with the Verilog HDL*, Prentice Hall India, 2nd Edition, 2010.
3. Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, Prentice Hall Professional, 2003.

CS 312

Microcontrollers for Embedded Systems

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT–I

Introduction to Microcontrollers, Embedded Systems, 8051-Architecture: instruction set, addressing modes and programming using 8051 microcontroller.
Memory Organization, Program Memory, Data Memory, Interrupts, Peripherals: Timers, Serial Port, I/O Port, Addressing Modes, Instruction Set, Programming

UNIT – II

Comparison of various families of 8-bit micro controllers, Interfacing of LCD, ADC, DAC, Sensors and Keyboard using Microcontrollers , USB and RS232.

UNIT – III

Introduction: RISC/ARM Design Philosophy and Functional Block Diagram. Programmers Model: Data Types, Processor modes, Registers, General Purpose Registers, Program Status Register, CP15 Coprocessor, Memory and memory mapped I/O, Pipeline, Exceptions, Interrupts and Vector table, and ARM Processor Families.

UNIT – IV

ARM9 Microcontroller Architecture: Block Diagram, Features, Memory Mapping, Memory Controller (MC), External Bus Interface (EBI), Connections to Memory Devices System Timer (ST): Period Interval Timer (PIT), Watchdog Timer (WDT), Real- time Timer (RTT), Real Time Clock (RTC), and Parallel Input/Output Controller (PIO).

UNIT- V

Universal Synchronous Asynchronous Receiver Transceiver (USART): Block Diagram, Functional Description, Synchronous and Asynchronous Modes. Development & Debugging Tools for Microcontroller based Embedded Systems: Software and Hardware tools like Cross Assembler, Compiler, Debugger, and Simulator.

Suggested Reading:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, "The 8051 Microcontroller and Embedded Systems using Assembly and C", 2nd Edition, Prentice Hall
2. David Seal "ARM Architecture Reference Manual", 2001 Addison Wesley, England, Morgan Kaufmann Publishers.

CS 313

Advanced Computer Architecture

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Measuring Performance and cost: Performance measurement, Enhancements to Uni processsor 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, Programming Vector processors.

UNIT-IV

Array Processors: parallel array processor model, memory organization, interconnection networks: performance measures, static and dynamic topologies.

UNIT-V

Multiprocessors and Multi computers: Multiprocessor models, Shared-memory and distributed memory architectures, memory organization, Cache Coherence and Synchronization Mechanisms, parallel computer, performance models.

Suggested Reading:

1. John L. Hennessey and David A. Patterson, *Computer Architecture, A Quantitative Approach*, Elsevier, 4th Edition, 2007.
2. Sajjan G. Shiva, *Advance Computer Architecture*, Taylor Series Group, CRC press, 2006.
3. Kai Hwang, *Advanced Computer Architecture*, Mc Graw Hill, 1999.

CS 314

Embedded Programming

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT- I

Embedded OS Fundamentals (Linux): Introduction, Operating System fundamentals, General and Unix OS Architecture Embedded Linux. Booting Process in Linux GNU Tools: gcc, Conditional Compilation, Preprocessor directives, Command line arguments, Make files

UNIT-II

Embedded C Programming, Review of data types - Scalar types-Primitive types-Enumerated types- Subranges, Structure types-character strings -arrays- Functions Introduction to Embedded C-Introduction, Data types Bit manipulation, Interfacing C with Assembly. Embedded programming issues - Reentrancy, Portability, Optimizing, and testing embedded C programs.

UNIT-III

Embedded Applications using Data structures , Linear data structures- Stacks and Queues Implementation of stacks and Queues- Linked List - Implementation of linked list, Sorting, Searching, Insertion and Deletion, and non-linear structures.

UNIT-IV

Introduction to Object Oriented Concepts, Core Java/Java, Core- Java buzzwords, Overview of Java programming, Data types, variables and arrays, Operators, and Control statements.

UNIT –V

Embedded Java - Understanding J2ME, CDC (Connected Device configuration), CLDC (Connected Limited device configuration), MIDP applications.

Introduction to Programming and App Inventor: Introducing App Inventor, Getting Hands-On with App, Working with Media: Displaying Images, Duplicating Blocks and Using Dropdowns, Sounds, Color Blocks, Layout Components, Input, Variables, and Calculations: The Text Box Component, Performing Calculations, Storing Data with Variables , Creating Blocks with Type blocking, Math Functions. Decision Blocks and Boolean: Introduction to Operators, and Control statements.

Suggested Reading:

1. Jones, M Tim, *GNU/Linux Application Programming*, 2nd Edition, Course Technology PTR, 2008.

2. Prasad K.V.K.K, *Embedded /Real-Time Systems: Concepts, Design and Programming*, Dreamtech Press, 2003.

3. Sing Li and Jonathan Knudsen, *Beginning J2ME-From Novice to Professional*, Dreamtech Press, 3rd Edition, 2000.

4. Herbert Schildt, *The Complete Reference Java2*, Tata Mc GrawHill, 9th Edition, 2014.

5. Tony Gaddis, Rebecca Halsey, *Starting Out with App Inventor for Android (1e)*

CS 315

Field programmable Gate Arrays

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to ASIC: Types of ASIC's, ASIC design flow, Economics of ASIC's, Programmable ASIC's: CPLD and FPGA. Commercially available CPLD's and FPGA's: XILINX, ALTERA, ACTEL. FPGA Design cycle, Implementation tools: Simulation and synthesis, and Programming technologies.

UNIT-II

FPGA logic cell for XILINX, ALTERA and ACTEL ACT, Technology trends, AC/DC IO Cells, clock and power inputs, FPGA interconnect: Routing resources, Elmore's constant, RC delay and parasitic capacitance FPGA design flow, and Low-level design entry.

UNIT-III

FPGA physical design, CAD tools, Power dissipation, FPGA Partitioning, Partitioning methods.Floor planning: Goals and objectives, I/O, Power and clock planning, and Floor Planning tools.

UNIT-IV

Placement: Goals and objectives, Placement algorithms: Min-cut based placement, Iterative Improvement, and simulated annealing.

Routing: Goals and objectives, Global routing methods, Back-annotation. Detailed Routing: Goals and objectives, Channel density, Segmented channel routing, Maze routing, Clock and power routing, Circuit extraction, and DRC.

UNIT-V

Verification and Testing: Verification, Logic simulation, Design validation, Timing verification. **Testing Concepts:** Failures, Mechanism and faults, Fault coverage, ATPG methods, Design for testability, Scan Path Design, Boundary Scan design, BIST Design guidelines, and Design of a Testing machine.

Suggested Reading:

1. Pak and Chan, Samiha Mourad, *Digital Design using Field Programmable Gate Arrays*, Pearson Education, 1st edition, 2009.
2. Michael John Sebastian Smith, *Application Specific Integrated Circuits*, Pearson Education Asia, 3rd edition 2001.
3. S. Brown, R.J.Francis, J.Rose, Z.G.Vranesic, *Field programmable Gate Array*, BSP, 2007.

CS 316 System On Chip Architecture

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to Processor Design, Abstraction in Hardware Design, Processor design Tradeoffs, Design for Low power consumption, ARM processor as System-On-Chip: Acorn RISC Machine – Architecture inheritance – ARM programming model – ARM development tools – 3 and 5 stage pipeline ARM Organization – ARM instruction execution and implementation – ARM Co-processor interface.

UNIT-II

ARM Assembly Language programming: ARM instruction types – data transfer, data processing and control flow instructions.ARM instruction set- Co-processor instructions. Architectural support for High-level language.

UNIT-III

Memory Hierarchy: Memory size and speed – On-chip memory Caches –Cache design- an example- memory management.

UNIT-IV

Architectural Support for System Development: Advanced Microcontroller bus Architecture- ARM memory interface – ARM reference peripheral specification – Hardware system prototyping tools.

UNIT-V

Architectural Support for Operating System: An introduction to Operating Systems – ARM system control coprocessor- CP15 protection unit registers – ARM protection unit.- ARM MMU registers – ARM MMU Architecture – Synchronization – Context Switching input and Output.

Suggested Reading:

1. Steve Furber, *ARM System – on Chip Architecture*, Addison Wesley, 2nd Edition, 2000.
2. Ricardo Reis, *Design of System on a Chip: Devices and Component*, Springer publishers, 1st Edition, 2004.
3. Prakash Rashinkar, Peter Paterson and Leena Singh L, *System on Chip Verification Methodologies and Techniques*, Kluwer Academic Publishers, 2001

CS 217

Optimization Techniques

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Use of optimization methods. Introduction to classical optimization techniques, motivation to the simplex method, simplex algorithm, sensitivity analysis.

UNIT-II

Search methods - Unrestricted search, exhaustive search, Fibonocci method, Golden section method, Direct search method, Random search methods, Univariate method, simplex method, Pattern search method.

UNIT-III

Descent methods, Gradient of function, steepest decent method, conjugate gradient method.

Characteristics of constrained problem, Direct methods, The complex method, cutting plane method.

UNIT-IV

Review of a global optimization techniques such as Monte Carlo method, Simulated annealing and Tunneling algorithm.

UNIT V

Generic algorithm - Selection process, Crossover, Mutation, Schema theorem, comparison between binary and floating point implementation.

Suggested Reading:

1. SS Rao, *Engineering Optimization: Theory and Practice*, John Wiley & Sons, 1996.
2. Zhigmiew Michelewicz, *Genetic Algorithms + Data Structures = Evaluation Programs*, Springer-Verlag Berlin Heidelberg, 1992.

CS 318

Product Design and Quality Management

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Product Design and Development, Development processes, Identifying customer needs, Establishing product specifications, Concept generation, Concept selection, Product architecture, and Industrial design.

UNIT- II

Product Design and Development Design for Manufacturing, Prototyping, Robust Design, Patents and Intellectual property, Product Development Economics, and Managing Product Development Projects.

UNIT-III

Total Quality Management Principles and Practices: Definition of quality, Customer satisfaction and Continuous improvement.

UNIT -IV

Total Quality Management, Tools and Techniques: Statistical Process Control, Quality Systems, Bench Marking.

UNIT -V

Total Quality Management , Quality Function Deployment, Product Liability, Failure Mode and Effect Analysis, Management Tools.

Suggested Reading:

1. T Dale H. Besterfield , *Total Quality Management*, Pearson Education Asia, Pearson Education India, 2nd Edition, 2011.
2. Karl T Ulrich & Steven D Eppinger, *Product Design & Development*; McGraw-Hill Education, 5th Edition, 2011.

CS 319

Design for Testability

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to Testing: Testing Philosophy, Role of Testing, Digital and Analog VLSI Testing, VLSI Technology Trends affecting Testing, Types of Testing, Fault Modeling: Defects, Errors and Faults, Functional Versus Structural Testing, Levels of Fault Models, Single Stuck-at Fault.

UNIT –II

Logic and Fault Simulation: Simulation for Design Verification and Test Evaluation, Modeling Circuits for Simulation, Algorithms for True-value Simulation, Algorithms for Fault Simulation, ATPG.

UNIT –III

Testability Measures: SCOAP Controllability and Observability, High Level Testability Measures, Digital DFT and Scan Design: Ad-Hoc DFT Methods, Scan Design, Partial-Scan Design, Variations of Scan.

UNIT –IV

Built-In Self-Test: The Economic Case for BIST, Random Logic BIST: Definitions, BIST Process, Pattern Generation, Response Compaction, Built-In Logic Block Observers, Test-Per-Clock, Test-Per-Scan BIST Systems, Circular Self Test Path System, Memory BIST, Delay Fault BIST.

UNIT –V

Boundary Scan Standard: Motivation, System Configuration with Boundary Scan: TAP Controller and Port, Boundary Scan Test Instructions, Pin Constraints of the Standard, Boundary Scan Description Language: BDSL Description Components, Pin Descriptions.

Suggested Reading:

1. M. Abramovici, M.A.Breuer and A.D Friedman, *Digital Systems and Testable Design*, Wiley, 1994.
2. P.K. Lala, *Digital Circuits Testing and Testability*, Academic Press, 1997.

CS 320

DSP Architecture

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT I

Introduction to DSP Processors: Differences between DSP and other μp architectures, their comparison and need for special ASPs , RISC & CISC CPUs .

UNIT II

Overview of DSP processor design: fixed point DSPs – Architecture of TMS 320C 5X, C54X Processors, addressing modes, Assembly instructions, Pipelining and on-chip peripherals.

UNIT III

Floating point DSPs: Architecture of TMS 320 – IX- Data formats, Floating Point operations, addressing modes, instructions, pipelining and peripherals.

UNIT IV

DSP interfacing & software development tools: I/O interfacing with A/D converters, PCs, Dual port RAMs, EPGAs, DSP tools – Assembler, debugger, c-compiler, linker, editor, code composer studio.

UNIT V

Applications using DSP adaptive filtering, spectrum analysis, Echo cancellation modems, voice synthesis and recognition. Brief ideas of AD, Motorola DSP CPUs and their comparison with TI CPUs.

Suggested Reading:

1. C. Marren & G. Ewess, *A Simple Approach to Digital Signal Processing*, WILEY Inter-science, 1996.
2. K. Shin, *DSP Applications with TMS 320 Family*, Prentice Hall, 1987.
3. B. Ventakaramani, M. Bhaskar, *Digital Signal Processes, Architecture Processing and Applications*, Tata Mc Graw Hill, 2002.

CS 321

Graph Theory and its Applications

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT I

Preliminaries: Graphs, isomorphism, subgraphs, matrix representations, degree, operations on graphs, degree sequences

Connected graphs and shortest paths: Walks, trails, paths, connected graphs, distance, cut-vertices, cut-edges, blocks, connectivity, weighted graphs, shortest path algorithms T**rees:** Characterizations, number of trees, minimum spanning trees.

UNIT II

Special classes of graphs: Bipartite graphs, line graphs, chordal graphs
Eulerian graphs: Characterization, Fleury's algorithm, chinese-postman-problem

UNIT III

Hamilton graphs: Necessary conditions and sufficient conditions
Independent sets, coverings, matchings: Basic equations, matchings in bipartite graphs, perfect matchings, greedy and approximation algorithms

UNIT IV

Vertex colorings: Chromatic number and cliques, greedy coloring algorithm, coloring of chordal graphs, Brook's theorem
Edge colorings: Gupta-Vizing theorem, Class-1 graphs and class-2 graphs, equitable edge-coloring**.**

UNIT V

Planar graphs: Basic concepts, Eulers formula, polyhedrons and planar graphs, charactrizations,

planarity testing, 5-color-theorem

Directed graphs: Out-degree, in-degree, connectivity, orientation, Eulerian directed graphs, Hamilton directed graphs, tournaments

Suggested Reading:

1 F.Harry, Graph theory, Narosa Publications, 1988.
2.C.Berge: Graphs and Hypergraphs, North Holland/Elsevier, 1973
3. J A Bondy and U.S. R Murthy, *Graph Theory with Applications*, Elsevier Science Ltd, 1976.
4. Douglas B West, *Introduction to Graph Theory*, Prentice Hall, 2004

CS 111

MOBILE COMPUTING

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction: Wireless Transmission, Frequencies for Radio Transmission, Signals, Antennas, Signal Propagation, Multiplexing, Modulations, Spread Spectrum, MAC, SOMA, FDMA, TDMA, CDMA, Cellular Wireless Networks.

UNIT-II

Telecommunication Systems: GSM, GPRS, Satellite Networks, Basics, Parameters and Configurations, Capacity Allocation, FAMA and DAMA, Broadcast Systems, DAB, DVB, CDMA and 3G.

UNIT-III

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

UNIT-IV

Routing Ad-hoc Network Routing Protocols: Ad-hoc Network Routing Protocols, Destination Sequenced Distance Vector Algorithm, Cluster Based Gateway Switch Routing, Global State Routing, Fish-eye state Routing, Dynamic Source Routing, Ad-hoc on-demand Routing, Location Aided Routing, Zonal Routing Algorithm.

Mobile IP - Dynamic Host Configuration Protocol.

Traditional TCP - Classical TCP Improvements – WAP, WAP 2.0.

UNIT-V

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

File System Support for Mobility: Distributed File Sharing for Mobility support, Coda and other Storage Manager for Mobility Support.

Mobile Transaction and Commerce: Models for Mobile Transaction, Kangaroo and Joey transactions, Team Transaction, Recovery Model for Mobile Transactions, Electronic Payment and Protocols for Mobile Commerce.

Suggested Reading:

1. Jochen Schiller, *Mobile Communications*, Pearson Education, 2nd Edition, 2009.
2. Kurnkum Garg, *Mobile Computing*, Pearson Education , 2010
3. Asoke K Talukder, Roopa R Yavagal, *Mobile Computing*, TMH 2008.
4. Raj Kamal, *Mobile Computing*, Oxford, 2009.
5. "*A Survey of Mobile Transactions appeared in Distributed and Parallel databases*" 16,193-230, 2004, Kluwer Academics Publishers.
6. S. Acharya, M. Franklin and S. Zdonil, "*Balancing Push and Pull for Data Broadcast, Proceedings of the ACM SIGMOD*", Tuscon, AZ, May 1997.
7. S.Acharya, R. Alonso, M.Franklin and S.Zdonik, "*Broadcast Disks: Data Management for Assymetric Communication Environments, Proceedings of the ACM SIGMOD Conference*", San Jose, CA, May 1995.

CS 112

REAL TIME SYSTEMS

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction: Definition, Applications and Types of Real Time Systems, Typical Case Studies of Real Time Systems, Time Constraints.

A Reference Model for Real Time Systems: Processors and Resources, Periodic Task Model, Precedence and Data Dependency, Temporal, Foundational and Resource Parameters, Scheduling Hierarchy.

UNIT-II

Real Time Scheduling: Different Approaches- Clock Driven, Priority Driven, Scheduling of Periodic and Sporadic Jobs in Priority- Driven Systems.

UNIT-III

Resource Management Resources and Resource Access Control, Critical Section, Priority-Ceiling Protocols, concurrent Access to Data Objects.

UNIT-IV

Implementation Aspects: Timing Services and Scheduling Mechanisms, Other Basic Operating System Functions, Processor Reserves and Resource Kernel, Open System Architecture, Capabilities of Commercial Real Time Operating Systems, Predictability of General Purpose Operating Systems.

UNIT-V

Case Studies: Vx – Works, and RT Linux.

Suggested Reading:

1. Jane W.S. Liu, *Real Time Systems*, Pearson Education, 2001.
2. C.M. Krishna and Kang G. Shin, *Real Time Systems*, Mc-Graw Hill Companies Inc., 1997.
3. Raymond J.A. Buhr, Donald L. Bailey, *An Introduction to Real Time Systems*, Prentice Hall International, 1999.
4. K.V.K.K. Prasad, *Embedded Real Time Systems, Concepts, Design and Programming*, Dreamtech Press, 2003.

CS 114

MULTIMEDIA TECHNOLOGIES

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Media and Data Streams: Properties of multimedia systems, Data streams characteristics: Digital representation of audio, numeric instruments digital interface Bark concepts, Devices, Messages, Timing Standards Speech generation, analysis and transmission.

UNIT-II

Digital Image: Analysis, recognition, transmission, **Video**: Representation, Digitalization transmission **Animations**: Basic concepts, animation languages, animations control transmission

UNIT-III

Data Compression Standards: JPEG, H-261, MPEG DVI

Optical storage devices and Standards: WORHS, CDDA, CDROM, CDWO, CDMO.

Real Time Multimedia, Multimedia file System.

UNIT-IV

Multimedia Communication System: Collaborative computing session management, transport subsystem, QOS, resource management.

Multimedia Databases: Characteristics, data structures, operation, integration in a database model. **A Synchronization**: Issues, presentation requirements, reference to multimedia synchronization, MHEG

UNIT-V

Multimedia Application: Media preparation, Composition, integration communication, consumption, entertainment.

Suggested Reading:

1. Ralf Steninmetz, Klara Hahrstedt, *Multimedia: Computing, Communication and Applications,* PHI PTR Innovative Technology Series.
2. John F.Koegel Bufford, *Multimedia System*, Addison Wesley, 1994.
3. Mark Elsom – Cook, *Principles of Interactive Multimedia* , Tata Mc-Graw Hill, 2001.
4. Judith Jefcoate, *Multimedia in Practice: Technology and Application* , PHI 1998.

CS 115

DATA MINING

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

Introduction: Why Data Mining? What is Data Mining? What kinds of data can be mined? What kinds of patterns can be mined? Which technologies are used ? Which kinds of applications are Targeted? Major issues in Data Mining. Getting to know your data: Data objects and attributed types. Basic statistical descriptions of data. Data visualization, Measuring data similarity and dissimilarity.

UNIT-II

Mining frequent patterns, Associations and correlations, Basic concepts and methods, Basic concepts, Frequent Item set Mining Methods, Which patterns are interesting? Pattern evaluation methods.

UNIT-III

Classification : Basic concepts, Decision tree induction, Bayes classification methods,

Classification: Advance methods, Bayesian Belief Network, Classification by back propagation, Support vector machine,

UNIT-IV

Cluster Analysis: Concepts and Methods: Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Evaluation of clustering.

UNIT-V

Data Mining Trends and Research Frontiers, Mining Complex Data Types, Other Methodologies of Data Mining, Data Mining Applications, Data Mining and Society, Data Mining trends.

Suggested Reading:

1. Jiawei Han, Micheline Kamber, Jin Pei, Data Mining: Concepts & Techniques, 3rd Edition.,Morgon Koffman ,2011
2. Vikram Pudi P.Radha Krishna, *Data Mining*, Oxford University Press, 1st Edition, 2009.
3. Pang-Ning Tan, Michael Steinbach, Vipin kumar, *Introduction to Data Mining*, Pearson Education, 2008.

CS 116

NETWORK SECURITY

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction: Attributes of Security, Integrity, Authenticity, Non-repudiation, Confidentiality Authorization, Anonymity, Types of Attacks, DoS, IP Spoofing, Replay, Man-in-the-Middle attacks General Threats to Computer Network, Worms, Viruses, -Trojans

UNIT-II

Secret Key Cryptography : DES, Triple DES, AES, Key distribution, Attacks

Public Key Cryptography: RSA, ECC, Key Exchange (Diffie-Hellman), Java Cryptography Extensions, Attacks

UNIT-III

Integrity, Authentication and Non-Repudiation : Hash Function (MD5, SHA5), Message Authentication Code (MAC), Digital Signature (RSA, DSA Signatures), Biometric Authentication.

UNIT-IV

PKI Interface: Digital Certificates, Certifying Authorities, POP Key Interface, System Security using Firewalls and VPN's.

Smart Cards: Application Security using Smart Cards, Zero Knowledge Protocols and their use in Smart Cards, Attacks on Smart Cards

UNIT-V

Applications: Kerberos, Web Security Protocols (SSL), IPSec, Electronic Payments, E-cash, Secure Electronic Transaction (SET), Micro Payments, Case Studies of Enterprise Security (.NET and J2EE)

Suggested Reading:

1. William Stallings, *Cryptography and Network Security*, 4th Edition. Pearson,. 2009.
2. Behrouz A Forouzan, *Cryptography and Network Security*, TMH, 2009
3. Joseph Migga Kizza, *A Guide to Computer Network Security*, Springer, 2010
4. Dario Cataiano, *Contemporary Cryptology*, Springer, 2010.

CS 117 **MACHINE LEARNING**

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction: Learning, Types of Machine Learning.
Concept learning: Introduction, Version Spaces and the Candidate Elimination Algorithm. **Learning with Trees:** Constructing Decision Trees, CART, Classification Example

UNIT-II

Linear Discriminants: The Perceptron, Linear Separability, Linear Regression

Multilayer Perceptron (MLP): Going Forwards, Backwards, MLP in practices, Deriving back **Propagation SUPPORT Vector Machines**: Optimal Separation, Kernels

UNIT-III

Some Basic Statistics: Averages, Variance and Covariance, The Gaussian, The Bias-Variance Tradeoff Bayesian learning: Introduction, Bayes theorem. Bayes Optimal Classifier, Naive Bayes Classifier.

Graphical Models: Bayesian networks, Approximate Inference, Making Bayesian Networks, Hidden Markov Models, The Forward Algorithm.

UNIT-IV

Evolutionary Learning: Genetic Algorithms, Genetic Operators, Genetic Programming **Ensemble learning:** Boosting, Bagging

Dimensionality Reduction: Linear Discriminant Analysis, Principal Component Analysis

UNIT-V

Clustering: Introduction, Similarity and Distance Measures, Outliers, Hierarchical Methods, Partitional Algorithms, Clustering Large Databases, Clustering with Categorical Attributes, Comparison

Suggested Reading:

1. Tom M. Mitchell, *Machine Learning*, Mc Graw Hill, 1997
2. Stephen Marsland, *Machine Learning - An Algorithmic Perspective*, CRC Press, 2009
3. Margaret H Dunham, *Data Mining*, Pearson Edition., 2003.
4. Galit Shmueli, Nitin R Patel, Peter C Bruce, *Data Mining for Business Intelligence*, Wiley India Edition, 2007
5. Rajjan Shinghal, *Pattern Recognition*, Oxford University Press, 2006.

CS 121

CLOUD COMPUTING

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

Unit- I

Introduction, Benefits and challenges, Cloud computing services, Resource Virtualization, Resource pooling sharing and provisioning

Unit -II

Scaling in the Cloud, Capacity Planning , Load Balancing, File System and Storage,

Unit-III

Multi-tenant Software, Data in Cloud , Database Technology, Content Delivery Network, Security Reference Model , Security Issues, Privacy and Compliance Issues

Unit-IV

Portability and Interoperability Issues, Cloud Management and a Programming Model Case Study, Popular Cloud Services

Unit- V

Enterprise architecture and SOA, Enterprise Software , Enterprise Custom Applications, Workflow and Business Processes, Enterprise Analytics and Search, Enterprise Cloud Computing Ecosystem.

Suggested Reading:

1. Cloud Computing - Sandeep Bhowmik, Cambridge University Press, 2017.

2. Enterprise Cloud Computing - Technology, Architecture, Applications by Gautam Shroff, Cambridge University Press, 2016.

3. Kai Hwang, Geoffrey C.Fox, Jack J.Dongarra, "*Distributed and Cloud Computing From Parallel Processing to the Internet of Things*", Elsevier, 2012.

CS 122

SOFT COMPUTING

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to Soft Computing and Neural Networks: Evolution of Computing Soft Computing Constituents From Conventional AI to Computational Intelligence-Machine Learning Basics.

UNIT II

Genetic Algorithms: Introduction to Genetic Algorithms (GA) –Applications of GA in Machine Learning-Machine Learning Approach to Knowledge Acquisition.

UNIT III

Neural networks: Machine Learning Using Neural Network, Adaptive Networks –Feed forward Networks –Supervised Learning Neural Networks–Radial Basis Function Networks-Reinforcement Learning– Unsupervised Learning Neural Networks–Adaptive Resonance architectures – Advances in Neural networks.

UNIT IV

Fuzzy Logic: Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Membership Functions, Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems ,Fuzzy Expert Systems, Fuzzy Decision Making.

UNIT V

Neuro-Fuzzy Modeling: Adaptive Neuro, Fuzzy Inference Systems, Coactive Neuro, Fuzzy Modeling, Classification and Regression Trees, Data Clustering Algorithms, Rule base Structure Identification, Neuro-Fuzzy Control, Case studies.

Suggested Reading:

1. Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice- Hall of India, 2003.
2. George J. Klir and Bo Yuan, *Fuzzy Sets and Fuzzy Logic-Theory and Applications*, Prentice Hall, 1995.
3. James A. Freeman and David M. Skapura, *Neural Networks Algorithms, Applications, and Programming Techniques*, Pearson Edn., 2003.
4. Mitchell Melanie, *An Introduction to Genetic Algorithm*, Prentice Hall, 1998.
5. David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1997.

CS 123

ARTIFICIAL NEURAL NETWORKS

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

Unit-I

Background to ANN: Introduction to artificial neural networks (ANN), intelligence, learning and knowledge. Historical development of Artificial Intelligence (AI) leading to ANN. PDP models -- Interactive and competetion (IAC) and Constraint Satifaction (CS) models.

Unit-II

Baiscs of ANN: Basics of ANN, terminology, models of neurons, topology, basic learning laws, activation and synaptic dynamics models

Unit-III

Analysis of Feedforward Neural Networks (FFNN): Overview, linear associative networks, perceptron network, multilayer perceptron, gradient descent methods, backpropagation learning

Unit-IV

Analysis of Feedback Neural Networks (FBNN): Overview, Hopfield model, capacity, energy analysis, state transition diagrams, stochastic networks, Boltzmann-Gibbs Law, simulated annealing, Boltzmann machine

Unit-V

Applications of ANN: Travelling salesman problem, image smoothing, speech recognition and texture classification.

Suggested Reading:

1.B Yegnanarayana, Artificial Neural Networks, Prentice-Hall of India, New Delhi, 1999

2. Simon Haykin, Neural networks and learning machines, Pearson Education, 2011

3. Jacek M Zurada, Introduction to artificial neural systems, PWS publishing Company, 1992

4. David E Rumelhart, James McClelland, and the PDP research group, Eds, Parallel and Distributed Processing: Explorations in Microstructure of Cognition, Vol 1, Cambridge MA: MIT Press, 1986a

5. James McClelland, David E Rumelhart, and the PDP research group, Eds, Parallel and Distributed Processing: Explorations in Microstructure of Cognition, Vol 2, Cambridge MA: MIT Press, 1986b

6. David Rumelhart, James McClelland, and the PDP research group, Eds, Parallel and Distributed Processing: A handbook of models, Cambridge MA: MIT Press, 1989

CS 124

SOFTWARE PROJECT MANAGEMENT

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Conventional Software Management, Evolution of Software Economics, Improving Software Economics, Old Way & New.

UNIT-II

Life – Cycle phases, Artifacts of the process, Model Based Software Architectures, Workflows of the Process, Checkpoints of the process.

UNIT-III

Iterative Process Planning, Project Organizations & Responsibilities, Process Automation, Project Control of Process Instrumentation, Tailoring the Process.

UNIT-IV

Modern Project profiles, Next Generation Software Economics, Modern process Transitions, Managing Contacts, Managing People & Organizing Terms.

UNIT-V

Process improvement & mapping to the CMM, ISO 12207 – an overview, programme management.

Suggested Reading:

1. Walker Royce, *Software Project Management – A Unified frame work*, Pearson Education, Addision, 1998,
2. Bob Hughes and Mike Cotterell , *Software Project Management*, Tata Mc Graw Hill, 3rd Edition, 2010.
3. Watt.S. Humphery, *Managing Software Process* , Addison - Wesley, 2008.

CS 127

RELIABILITY AND FAULT TOLERANCE

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to Reliability Engineering: Reliability, Repairable and Non-repairable Systems, Maintainability and Availability, Designing, Reliability, Repairable and Non-repairable Systems, MTBF MTBF, MTTF MDT, k out of in systems.

UNIT-II

Software Reliability:Software Reliability, Software Reliability Vs Hardware Reliability, Failures and Faults, Classification of Failures, Counting, System configuration, Components and Operational Models, Concurrent Systems, Sequential Systems, Standby Redundant Systems.
Software Reliability Approaches: Fault Avoidance, Passive Fault Detection, Active Fault Detection, Fault Tolerance, Fault Recovery, Fault Treatment.

UNIT-III

Software Reliability Modeling: Introduction to Software Reliability Modeling, Parameter Determination and Estimation, Model Selection, Markovian Models, Finite and Infinite failure category Models, Comparison of Models, Calendar Time Modeling.

UNIT-IV

Fault Tolerant Computers: General Purpose Commercial Systems, Fault Tolerant Multiprocessor and VLSI based Communication Architecture.
Design – N – Version programming Recovery Block, Acceptance Tests, Fault Trees, Validation of Fault Tolerant Systems.

UNIT-V

Fault Types: Fault Detection and Containment, Redundancy, Data Diversity, Reversal, Reversal Checks, Obtaining Parameter Values, Reliability Models for Hardware Redundancy, Software Error Models, Checks, Fault /Tolerant Synchronization, Synchronization in Software.

Suggested Reading:

1. John D. Musa, *Software Reliability*, McGraw Hill, 1995.
2. Patrick O'Connor, *Practical Reliability Engineering*, 4th Edition, John Wesley & Sons, 2003.
3. C.M. Krishna, Kang G. Shin, *Real Time Systems*, McGraw Hill, 1997.

CS 131

Software Engineering for RTS

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

I**ntroduction:** Review of Software Engineering Concepts, Characteristics of Real Time Systems, Importance of including Time Factor, The Real Time System Life Cycle: Requirement Specifications, State Charts.

UNIT-II

Structured Design Approaches: Event Based Model, Process-Based Structured Design, Graph-Based Theoretical Model, Petri Net Models: Stochastic Petri Net (SPN) Model Analysis, Annotated Petri Nets, Time-Augmented Petri Nets, Assessment of Petri Net Methods.

UNIT-III

Axiomatic Approaches: Weakest Precondition Analysis, Real Time Logic, Time Related History variables, State Machines and Real-Time Temporal Logic.

UNIT-IV

Language Support Restrictions: Real-Time Programming Descipline, Real-Time Programming Languages, Schedulability Analysis.

UNIT-V

Verification and Validation of Real-Time Software: Testing Real Time Properties, Simulation as Verification Tool, Testing Control and Data Flow, Proof Systems, Operational Approach.

Suggested Reading:

1. Shem – Tow Levi and Ashok K. Agarwal, *Real Time System Design*, McGraw Hill International Editions, 1999.
2. Cooling J.E. Jim Cooling, *Software Engineering for Real Time Systems*, Addison Wesly,2002

CS 132

Simulation and Modelling

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to simulation: Advantages & Dis-advantages of simulation – Areas of applications, Systems and Systems Environment, Concept of a system, Discrete & Continuous system – Models, types of models, Steps in a simulation study – Examples, Discrete – Event System simulation.

UNIT-II

Overview of Statistical Models and Queuing Systems, Programming languages for Simulation: Continuous and Discrete Simulation Languages – GPSS, SIMAN, SIMSCRIPT, MATLAB and SIMULINK

.

UNIT-III

Random Numbers: Generation, Properties of Random Numbers, Generation of Pseudo Random Numbers, Tests for Random Numbers.

Random Variate: Generation, Inverse Transformation Technique, Uniform Distribution, Exponential Distribution, Weibul's Distribution, Triangular Distribution, Empirical Continuous Distribution, Discrete Distributions, Direct Transformation for the Normal Distribution, Convolution Method of Erlang Distribution, Acceptance Rejection Techniques: Poisson Distribution, Gamma Distribution.

UNIT-IV

Input Data Analysis: Data Collection: Identify the Distribution, Parameter and Estimation.

Goodness of fit tests: Chi-Square Test – KS Test; Multivariate and time series input models, Verification and Validations of Simulation Models, Model Building, Verification and Validation: Verification of Simulation Models, Calibration and Validation of Models, face validity, Validation of Model Assumptions. Validation Input/output Transformations, Input/output Validation using Historical Input Data, Input/output Validation Sing Turning Test.

UNIT-V

Output Data Analysis, Stochastic, Nature of output data, Types of Simulation with respect to output Analysis, Measures of Performance and their Estimation, output Analysis for Terminating Simulations, Output Analysis for steady – State Simulations.

Comparison and Evaluation of Alternative System Designs: Comparison of several system Designs, Statistical Models for Estimating the Effect of Design Alternatives.

Suggested Reading:

1. Jabey Banks, John S. Cansen and Barry L. Nelson, *Discrete – Event System Simulation*, Prentice Hall of India, 2001.
2. Nursing Deo, *System Simulation with Digital computer*, Prentice Hall of India, 1979.
3. Anerill M. Law and W. David Kelton, *Simulation Modelling and Analysis*, McGraw Hill. 2001.
4. Agam kumar tyagi, MATLAB and Simulink for Engineers, Oxford Publishers, 2011

CS 133

ADVANCED OPERATING SYSTEMS

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Architecture of Distributed Systems: Types, Distributed Operating System, Issues in Distributed Operating Systems, Theoretical Foundations: Global Clock, Lamport's Logical Clock, Vector Clocks, Global State, and Termination Detection.

UNIT-II

Distributed Mutual Exclusion: Classification, requirement, performance, non-token based algorithms, Lamport's algorithm, the Richart-Agarwala algorithm, token-based algorithm-Suzuki liasamil's broadcast algorithm, Singhals heuristic algorithm.

Deadlock Detection: Resource Vs Communication deadlock, A graph- theoretic model, prevention, avoidance, detection, control organization, centralized deadlock-detection algorithm, the completely centralized algorithm, the HO-Ramamoorthy algorithm. Distributed deadlock detection algorithm - path - pushing, edge-chasing, hierarchical deadlock detection algorithm, menace-muntz and Ho-Ramamoorthy algorithm. Agreement Protocols: The system model, the Byzantine agreement, and the consensus problem.

UNIT-III

Distributed File System: Mechanisms, Design Issues.

Case Studies: Sun NFS, Sprite File System, DOMAIN, Coda File System.

Distributed Shared Memory: Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues.

Case Studies: IVY, Mirage, Clouds.

Distributed Scheduling : Issues in Load Distribution, Components of Algorithm, Stability Load Distributing Algorithm, Performance.

UNIT IV

Failure Recovery: Backward, Forward Error Recovery in Concurrent Systems, Consistent Set of Check Points, Synchronous and Asynchronous Check Pointing and Recovery.

Fault Tolerance: Commit Protocols, Non-Blocking Commit Protocols, Voting Protocols.

Protection and Security: Access Matrix, Private Key, Public key, and Kerberos System.

UNIT -V

Multiprocessor Operating Systems: Motivation, Basic Multiprocessor System Architecture, Interconnection Networks for Multiprocessor Systems, Caching, Hypercube Architecture. Threads, Process Synchronization, Processor Scheduling, and Memory Management.

Database Operating System: Concurrence Control, Distributed Databases, and Concurrency Control Algorithms.

Suggested Reading:

1. Singhal M, Shivaratri N.G, *Advanced Concepts in Operating Systems*, McGraw-Hill Intl., 1994.
2. Pradeep K Sinha, *Distributed Operating Systems Concepts and Design*, PHI, First Edition, 2002.
3 Andrew S. Tanenbaum, *Distributed Operating Systems*, Pearson Education India, First Edition, 2011

CS 134

OBJECT ORIENTED SOFTWARE ENGINEERING

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Information Systems: Problems in Information systems Development, Project life cycles, Managing Information System Development, User Involvement and Methodological Approaches, Basic Concepts and Origins of Object Orientation Modeling Concepts.

UNIT-II

Requirement Capture, User Requirements, Requirements Capture and Modelling, Requirement Analysis, Use Case Realization, The Class Diagram, Assembling the Analysis Class Diagram, Refining the Requirement Models, Component-based Development, Software Development Patterns, Object Interaction, Object Interaction and Collaboration, Interaction Sequence Diagrams, Collaboration Diagrams, Model Consistency

UNIT-III

Specifying Operations, The Role of Operation Specifications, Contracts, Describing Operation Logic, Object Constraint Language, Creating an Operation Specification, Specifying Control, States and Events, Basic Notation, Further Notation, Preparing a Statechart, Consistency Checking, Quality Guidelines, Moving Into Design, Logical and Physical Design, System Design and Detailed Design, Qualities and Objectives of Analysis and Design, Measurable Objectives in Design, Planning for Design, System Design, The Major Elements of System Design, Software Architecture, Concurrency, Processor Allocation, Data Management Issues, Development Standards, Prioritizing Design Trade-offs, Design for Implementation

UNIT-IV

Object design, Class Specification, Interfaces, Criteria for Good Design, Designing Associations, Integrity Constraints, Designing Operations, Normalization, Design Patterns, Software Development Patterns, Documenting Patterns-Pattern Templates, Design Patterns, How to use Design Patterns, Benefits and Dangers of Using Patterns, Human Computer Interaction, The User Interface, Approaches to User Interface Design, Standards and Legal Requirements, Designing Boundary Classes, The Architecture of the Presentation Layer, Prototyping the User Interface, Designing Classes, Designing Interaction with Sequence Diagrams, The Class Diagram Revisited, User Interface Design Patterns, Modelling the Interface Using Statecharts.

.

UNIT-V

Data Management Design, Persistence, File Systems, Database Management Systems, Designing for Relational Database Management Systems, Designing for Object Database Management Systems, Distributed Databases, Designing Data Management Classes, Implementation, Software Implementation, Component Diagrams, Deployment Diagrams, Software Testing, Data Conversion, User Documentation and Training, Implementation Strategies, Review and Maintenance, Reusable Components, Planning a Strategy for Reuse, Commercially Available Componentware, Managing Object Oriented Projects, Resource Allocation and Planning, Managing Iteration, Dynamic Systems Development Method, Extreme Programming, Software Metrics, Process Patterns, Legacy Systems, System Development Methodologies, 'Method' and 'Methodology', A Brief Historical Review, The Unified Software Development Process, Participative Design Approaches, Issues in Choosing a Methodology, Hard versus Soft Methodologies.

Suggested Reading:

1. Simon Benett, Steve McRobb and Ray Farmer, *Object Oriented System Analysis and Design using UML*, McGraw-Hill Education, 2010.
2. Grady Booch, James Rumbaugh, Ivar Jacobson, *The Unified Modeling language-User guide*, Pearson Education India, 2nd Edition, 2005.
3. Subhash Mehta, Suresh K. Basandra, *Object Oriented Software Engineering*, Galgotia, 2004.

CS 135

DISTRIBUTED COMPUTING

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT -I

Introduction: Definition of Distributed Systems, Goals: Connecting Users and Resources, Transparency, Openness, Scalability, Hardware Concepts: Multiprocessors, Homogeneous Multicomputer systems, Heterogeneous Multicomputer systems, Software Concepts: Distributed Operating Systems, Network Operating Systems, Middleware, The client-server model: Clients and Servers, Application Layering, Client-Server Architectures.

UNIT II

Communication: Layered Protocols, Lower-Level Protocols, Transport Protocols, Higher-Level Protocols, Remote Procedure Call: Basic RPC Operation, Parameter Passing, Extended RPC Models, Remote Object Invocation: Distributed Objects, Binding a Client to an Object; Static verses Dynamic Remote Method Invocations, Parameter Passing, Message Oriented Communication: Persistence and synchronicity in Communication, Message-Oriented Transient Communication, Message-Oriented' Persistent Communication, Stream Oriented Communication: Support for Continuous Media, Streams and Quality of Service, Stream Synchronization.

UNIT -III

Process: Threads: Introduction to Threads, Threads in Distributed Systems, Clients: user Interface-:, Client-Side Software for Distribution Transparency, Servers: General Design Issues, Object Servers, Software Agents: Software Agents in Distributed Systems, Agent Technology, Naming: Naming Entities: Names, Identifiers, and Address, Name Resolution, The Implementation of a Name System, Locating Mobile Entities: Naming verses Locating Entities, Simple Solutions, Home-Based Approaches, Hierarchical Approaches.

UNIT -IV

Distributed Object Based Systems: CORBA: Overview of CORBA, Communication, Processes, Naming, Synchronization, Caching and Replication, Fault Tolerance, Security, Distributed COM: Overview of DCOM, Communication, Processes, Naming, Synchronization, Replication, Fault Tolerance, Security, GLOBE: Overview of GLOBE, Communication, Process, Naming, Synchronization, Replication, Fault Tolerance, Security, Comparison of CORBA, IDCOM, and

Globe: Philosophy, Communication, Processes, Naming, Synchronization, Caching and Replication Fault Tolerance, Security, MTN

UNIT-V

Distributed Multimedia Systems: Introduction, Characteristics of Multimedia Data, Quality of Service Management: Quality of Service negotiation, Admission Control, Resource Management Resource Scheduling.

Suggested Reading:

1. Andrew S. Tanenbaum and Marteen Van Steen, *Distributed System*s: *Principles and Paradigms,* Pearson Prentice Hall, 2nd Edition, 2010.
2. Colouris G., Dollimore Jean, Kindberg Tim, *Distributed Systems Concepts and Design*, 3rd Edition Pearson Education, 5th Edition, 2011.

CS 136

ADVANCED DATABASES

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Object Based Databases: Overview, Complex Data Types, Structured Types and Inheritance in SQL, Table Inheritance, Array and Multi-set. Types in SQL, Object-Identity and Reference Types in SQL, Implementing O-R features, Persistent Programming Languages, Object-Relational Mapping, Object-Oriented versus Object-Relational.

UNIT-II

X M L: Motivation, Structure of XML data, XML Document Schema, Querying and Transformation, Application Program Interface to XML, Storage of XML data, XML applications.

UNIT-III

Query Processing : Overview, Measures of Query Cost, Selection Operation, Sorting, join Operation, Other Operations, Evaluation of Expressions.

Query Optimization: Overview, Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans, Materialized Views.

UNIT-IV

Parallel Databases: Introduction,1/0 Parallelism, Interquery Parallelism, Intraquery Parallelism, Intra-operation Parallelism, Interoperation Parallelism, Query Optimization, Design of Parallel Systems.

Distributed Databases: Homogeneous and Heterogeneous Database, Distributed Data Storage, Distributed. Transactions, Commit Protocols, Concurrency Control in Distributed Databases, Availability, Distributed Query Processing, Heterogeneous Distributed Databases, Cloud-Based Databases, Directory Systems.

UNIT- V

Advanced Application Development: Performance Tuning, Performance Benchmarks Other Issues in Application Development, Standardization.

Spatial and Temporal Data and Mobility: Motivation, Time in Databases, Spatial and Geographic Data, Multimedia Databases, Mobility and Personal Databases.

Suggested Reading:

1. Abraham Silberschatz, Henry F Korth, S Sudarshan, *Database System Concepts*, McGrawHill International Edition, 6th Edition, 2010.

2. Elmasri Navathe, Somayajulu, Gupta , *Fundamentals of Database Systems*, Pearson Education, 4th Edition, 2006.

3. CJ Date, A Kannan, S Swamynathan, *An Introduction to Database Systems*, Pearson Education, 8th Edition, 2006.

4. Raghu Ramakrishnan, and Johannes Gehrke, *Database Managemen*t *Systems*, McGraw-Hill International Edition, 3rd Edition, 2002.

CS 211

Parallel Algorithms

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to Parallel Algorithms and Architectures: Approaches to Design of Parallel Algorithms, Architectural Constraints and Design and Analysis of Parallel Algorithms, Performance Measures of Parallel Algorithms

UNIT-II

Parallel Design Strategies: Parallel Prefix. Computations, Pointer Jumping, Matrix Operations in Parallel.

Dense Matrix algorithms: Matrix vector Multiplication and Matrix- matrix multiplication

UNIT-III

Parallel Sorting: Issues in Sorting on Parallel Computers, Sorting Networks, Bubble Sort and its Variants, Quicksort, Bucket and Sample Sort.

UNIT-IV

Parallel Graph Algorithms: Definitions and Representations, Minimum Spanning Tree: Prim's Algorithm, Single Source Shortest Path - Dijkstra's Algorithm, All pairs shortest path algorithms, Algorithms for Sparse Graphs.

UNIT-V

Search Algorithms for Discrete Optimization Problems: Definitions, Sequential search Algorithms, Search Overhead Factor, Parallel Depth first Search, Parallel Breadth first Search, Speedup factors in Parallel Search Algorithms.

Suggested Reading:

I. Kenneth A. Berman and Jerome Paul, *Parallel Algorithms* , Cengage Learning, 2002.

2. Ananth Grama and Anshul Gupta, *Introduction to Parallel Computing*, Pearson Education Second Edition, 2004.

CS 212

Grid Computing

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT-I

Introduction to Grid Computing: Grid Computing Concept, History of Distributed Computing, Computational Grid Applications, Grid Computing Infrastructure Development, Grid Computing Software Interface.

Job Submission: Introduction, Globus Job Submission. Transferring Files.

UNIT-II

Schedulers: Scheduler Features, Scheduler Examples, Grid Computing Meta-Schedulers Distributed Resource Management Application (DRMAA).

Security Concepts: Introduction, Symmetric Key Cryptography, Asymmetric Key Cryptography (Public Key Cryptography), Public Key Infrastructure. Systems/Protocols Using Security Mechanisms.

Grid Security: Introduction, Grid Security Infrastructure (GSI), Delegation, Higher-Level Authorization Tools.

UNIT-III

System Infrastructure I: Web Services: Service-Oriented Architecture, Web Services and Web Service Implementation.

System Infrastructure II: Grid Computing Services: Grid Computing and Standardization Bodies, Interacting Grid Computing Components, Open Grid Services Architecture (OGSA), WSRF.

User-Friendly Interfaces: Introduction, Grid Computing Workflow Editors, Grid Portals.

UNIT-IV

Grid-Enabling Applications: Introduction, Parameter Sweep, Using an Existing Program on Multiple Grid Computers, Writing an Application Specifically for a Grid, Using Multiple Grid Computers to Solve a Single Problem.

UNIT-V

Case Studies: Globus-Overview of Globus Toolkit 4, Installation of Globus, GT4 Configuration, Main Components and programming Model using Globus.

gLite: Introduction, Internal Workings of gLite, Logging and Bookkeeping (LB), Security Mechanism Using gLite, Resource management using Gridway and Gridbus Scheduling using Condor, SGE, PBS, LSF Grid scheduling with QoS.

Suggested Reading:

1. Barry Wilkinson, *Grid Computing Techniques and Applications*, CRC Press, 2010.
2. Frederic Magoules, Jie Pan, Kiatan Tan, Abhinit Kumar , *Introduction to Grid Computing*, CRC Press, 2009.
3. Vladimir Silva, *Grid Computing for Developers*, Dreamtech Press, 2006.
4. Ian Foster, and Carl Kesselman, *The Grid 2: Blueprint for a new computing Infrastructure*, Elsevier Series, 2004
5. Fran Berman, Geoffrey Fox, Anthony J.G Hey, *Grid Computing: Making the Global Infrastructure a Reality*, Wiley Publishers, 2003.
6. Joshey Joseph, Craig Fellenstein, *Grid Computing*, IBM Press, 2004.

CS 213

Real Time Operating Systems

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT I

Brief Review of Unix Operating Systems (Unix Kernel – File system, Concepts of – Process, Concurrent Execution & Interrupts. Process Management – forks & execution. Programming with system calls, Process Scheduling. Shell programming and filters).

Portable Operating System Interface (POSIX) – IEEE Standard 1003.13 & POSIX real time profile. POSIX versus traditional Unix signals, overheads and timing predictability.

UNIT II

Hard versus Soft Real-time systems – examples, Jobs & Processors, Hard and Soft timing constraints, Hard Real-time systems, Soft Real-time systems. Classical Uniprocessor Scheduling Algorithms – RMS, Preemptive EDF, Allowing for Preemptive and Exclusion Condition.

UNIT III

Concept of Embedded Operating Systems, Differences between Traditional OS and RTOS. Real-time System Concepts, RTOS Kernel & Issues in Multitasking – Task Assignment, Task Priorities, Scheduling, Intertask Communication & Synchronization – Definition of Context Switching, Foreground ISRs and Background Tasks. Critical Section – Reentrant Functions, Interprocess Communication (IPC) – IPC through Semaphores, Mutex, Mailboxes, Message Queues or Pipes and Event Flags.

UNIT IV

VxWorks – POSIX Real Time Extensions, timeout features, Task Creation, Semaphores (Binary, Counting), Mutex, Mailbox, Message Queues, Memory Management – Virtual to Physical Address Mapping.

UNIT V

Debugging Tools and Cross Development Environment – Software Logic Analyzers, ICEs.
Comparison of RTOS – VxWorks, µC/OS-II and RT Linux for Embedded Applications.

Suggested Reading:

1. Jane W.S.Liu , *Real Time Systems* , Pearson Education, Asia, 2001.
2. Betcnhof, D.R., *Programming with POSIX threads*, Addison - Wesley Longman, 1997.
3. *VxWorks Programmers Guide*, Windriver, 1999.
4. Jean.J.Labrosse, *MicroC/OS-II*, Taylor & Francis, 2002.
5. C.M.Krishna and G.Shin, *Real Time Systems*, McGraw-Hill International Edition, 1997.

CS 214

Scripting Languages for Design Automation

Credits: 3:

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT I

Introduction to Python Programming: Program Development Cycle, Input, Processing, and Output, Variables, Performing Calculations (Operators, Type conversions, Expressions),

Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables.

Repetition Structures: Introduction, while loop, for loop, Input Validation Loops, Nested Loops.

UNIT II

Functions: Introduction, Defining and Calling a Void Function, Designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning Functions.

Lists and Tuples: Sequences, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List Methods and Useful Built-in Functions, Copying Lists, Processing Lists, Two-Dimensional Lists, Tuples.

UNIT III

File and Exceptions: Introduction to File Input and Output, Using Loops to Process Files, Processing Records, Exceptions.

Strings: Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings

Dictionaries and Sets: Dictionaries, Sets, Serializing Objects.

UNIT IV

Recursion: Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms.

Object-Oriented Programming: Procedural and Object-Oriented Programming, Classes, Working with Instances, Techniques for Designing Classes, Inheritance, Polymorphism.

UNIT V

GUI Programming: Graphical User Interfaces, Using the tkinter Module, Display text with Label Widgets, Organizing Widgets with Frames, Button Widgets and Info Dialog Boxes, Getting Input with Entry Widget, Using Labels as Output Fields, Radio Buttons, Check Buttons.

Suggested Reading:

1. Tony Gaddis, *Starting out With Python* , Pearson College Division, 3rd Edition, 2014.
2. John V Guttag, *Introduction to Computation and Programming using Python,* MIT Press, 3rd Edition, 2016.

CS 201

Parallel Computer Architecture

Credits: 3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT I

Instruction Level Parallelism: Concepts and challenges, Instruction Pipeline Design, Hardware and software approaches, Dynamic scheduling, Speculation, Compiler techniques for exposing ILP, Branch Handling Techniques.

UNIT-II

Advanced Processor Technologies: CISC and RISC Architectures, Superscalar Processors, and VLIW Architectures.

Memory Hierarchy Design: Cache basics and Cache performance, Reducing miss rate and Miss penalty, Multilevel cache hierarchies, Main memory organizations, and Design of Memory Hierarchies.

UNIT-III

Parallel Computer Models: Classification of Parallel Computers, Multiprocessors and Multicomputer, and Multi-vector and SIMD computers.

Shared Memory Multiprocessors: Cache Coherence, Memory Consistency, Snoopy-based Cache coherence protocols (MSI, MESI, MOESI).

UNIT-IV

Snoopy-based Multi-Processor Design: Single-level Caches with an Atomic Bus, Multi-level Cache Hierarchies, and Split-Transaction Bus.

Directory-Based Cache Coherence: Scalable Cache Coherence, Overview of Directory-based approaches, Design Challenges for Directory Protocols, Memory-Based Directory Protocols, Cache-Based Directory Protocols.

UNIT -V

Interconnection Network Design: Basic Definitions, Basic Communication Performance, Organizational Structure, Interconnection Topologies, Routing, Switch Design, and Flow Control.

Latency Tolerance: Overview of Latency Tolerance, Latency Tolerance in Explicit Message Passing, Latency Tolerance in a Shared Address Space - Block Data Transfer, Proceeding Past Long-Latency Events, Pre communication in a Shared Address Space, and Multithreading.

Suggested Reading:

1. John L. Hennessy, David A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers Inc., 5th Edition, 2012.
2. Id. Culler, Jaswinder Pal Singh, and Anoop Gupta, *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann, 1999.
3. Kai Hwang, *Advanced Computer Architecture*, Tata McGraw-Hill Education, 2nd Edition, 2011.

CS 202

Parallel Programming

Credits:3

Instruction : 3L hrs per week *Duration of SEE : 3 hours*

CIE : 30 Marks *SEE : 70 Marks*

UNIT -I

Principles of Parallel Algorithm Design - Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models.

UNIT-II

Communication Operations - One-to-All Broadcast and All-to-one Reduction, All-to-all Broadcast and Reduction, All-Reduce and Prefix-sum Operations, All-to-all Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operations.

UNIT-III

Analytical Modeling of Parallel Programs - Sources of Overhead in Parallel Programs, Performance Metrics for Parallel Systems, The Effect of Granularity on Performance, Scalability of Parallel Systems, Minimum Execution Time and Minimum Cost-Optimal Execution Time,

Asymptotic Analysis of Parallel Programs: Sorting and Graph Algorithms, Search algorithms for discrete optimization problems.

.

UNIT-IV

Introduction to Parallel Programming: Introduction to Parallel Programming, Introduction to OpenCL, OpenCL Device Architectures, Basic OpenCL Examples, Parallel programming using OpenCL/C++ AMP/CUDA.

UNIT-V

Introduction to OpenCL: Understanding OpenCL's Concurrency and Execution Model, Dissecting a CPU/GPU OpenCL Implementation.

Suggested Reading:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing* , 2nd Edition, Pearson Publishers.
2. David Kaeli, Perhaad Mistry, Dana Schaa and Dong Ping Zhang **,** *Heterogeneous Computing with OpenCL 2.0***,** 1st Edition, Mourgan Kaufmann, 2015.
3. Benedict Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, and Dana Schaa, *Heterogeneous Computing with OpenCL™ 1.2***,** Mourgan Kaufmann , 2011
4. Gregory V. Wilson, *Practical Parallel Programming*, PHI, 1998.

CS 103 **ADVANCED ALGORITHMS**

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

COURSE OBJECTIVE

- Introduce students to the advanced methods of designing and analyzing algorithms.
- The student should be able to choose appropriate algorithms and use it for a specific problem.
- To familiarize students with basic paradigms and data structures used to solve advanced algorithmic problems.
- Students should be able to understand different classes of problems concerning their computation difficulties.
- To introduce the students to recent developments in the area of algorithmic design.

COURSE OUTCOMES

- After completion of course, students would be able to:
- Analyze the complexity/performance of different algorithms.
- Determine the appropriate data structure for solving a particular set of problems.
- Categorize the different problems in various classes according to their complexity.
- Students should have an insight of recent activities in the field of the advanced data structure.

Unit1

Sorting: Review of various sorting algorithms, topological sorting,

Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkasra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.

Unit 2

Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST.

Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.

Unit 3

Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.

Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition.

Unit 4

Shortest Path in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming.

Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem.

Discrete Fourier Transform (DFT): In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm

Unit 5

Linear Programming: Geometry of the feasibility region and Simplex algorithm

NP-completeness: Examples, proof of NP-hardness and NP-completeness.

Approximation algorithms, Randomized Algorithms, Interior Point Method, Advanced Number Theoretic Algorithm. Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures

Suggested Reading:

1. "Introduction to Algorithms" byCormen, Leiserson, Rivest, Stein, 4th edition, McGraw Hill,
2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman.
3. "Algorithm Design" by Kleinberg and Tardos.

CS 104 **ARTIFICIAL INTELLIGENCE**

Credits: 3

Instruction: (3L) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

UNIT - 1

Introduction: History Intelligent Systems, Foundations of Artificial Intelligence, Sub areas of Al, Applications.

Problem Solving - State - Space Search and Control Strategies: Introduction, General Problem Solving Characteristics of problem, Exhaustive Searches, Heuristic Search Techniques, Iterative - Deepening A*, Constraint Satisfaction.

Game Playing, Bounded Look - ahead Strategy and use of Evaluation Functions, Alpha Beta Pruning.

UNIT – II

Logic Concepts and Logic Programming: Introduction, Propositional Calculus Propositional Logic, Natural Deduction System, Axiomatic System, Semantic Table, A System in Propositional Logic, Resolution, Refutation in Propositional Logic, Predicate Logic, Logic Programming.

Knowledge Representation: Introduction, Approaches to knowledge Representation, Knowledge Representation using Semantic Network, Extended Semantic Networks for KR, Knowledge Representation using Frames.

UNIT - III

Expert System and Applications: Introduction, Phases in Building Expert Systems Expert System Architecture, Expert Systems Vs Traditional Systems, Truth Maintenance Systems, Application of Expert Systems, List of Shells and tools.

Uncertainity Measure - Probability Theory: Introduction, Probability Theory, Bayesian Belief Networks, Certainity Factor Theory, Dempster - Shafer Theory.

UNIT - IV

Machine - Learning Paradigms: Introduction, Machine learning System, Supervised and Unsupervised Learning, Inductive Learning, Learning Decision Trees, Deductive Learning, Clustering, Support Vector Machines.

Artificial Neural Networks: Introduction Artificial Neural Networks, Single - Layer Feed Forward Networks, Multi - Layer Feed Forward Networks, Radial - Basis Function Networks, Design Issues of Artificial Neural Networks, Recurrent Networks

UNIT - V

Advanced Knowledge Representation Techniques: Case Grammars, Semantic Web.

Natural Language Processing: Introduction, Sentence Analysis Phases, Grammars and Parsers, Types of Parsers, Semantic Analysis, Universal Networking Knowledge.

Suggested Reading:

1. Saroj Kaushik, *Artificial Intelligence*, Cengage Learning India, First Edition, 2011.
2. Russell, Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, 2nd Edition, 2004.
3. Rich, Knight, Nair , *Artificial Intelligence*, Tata McGraw Hill, 3rd Edition 2009.