

UNIVERSITY DEPARTMENTS
ANNA UNIVERSITY CHENNAI :: CHENNAI 600 025
REGULATIONS – 2008
CURRICULUM FROM III & IV SEMESTERS FOR
B.TECH. INFORMATION TECHNOLOGY

SEMESTER III

CODE NO	COURSE TITLE	L	T	P	C
THEORY					
MA 9211	Mathematics III	3	1	0	4
EC 9212	Communication Techniques	3	0	0	3
CS 9202	Database Management Systems	3	0	0	3
IT 9201	Computer Organization	3	1	0	4
CS 9203	Programming and Data Structures II	3	0	0	3
CS 9201	Design and Analysis of Algorithms	3	0	0	3
PRACTICAL					
CS 9205	Database Management Systems Laboratory	0	0	3	2
CS 9206	Programming and Data Structures Laboratory II	0	0	3	2
CS 9207	Algorithms Laboratory	0	0	3	2
TOTAL		18	2	9	26

SEMESTER IV

CODE NO	COURSE TITLE	L	T	P	C
THEORY					
MA 9265	Discrete Mathematics	3	1	0	4
IT 9251	Formal Languages and Automata	3	0	0	3
IT 9252	Embedded Systems	3	0	0	3
CS 9252	Operating Systems	3	0	0	3
CS 9253	Web Technology	3	0	0	3
CS 9254	Software Engineering	3	0	0	3
PRACTICAL					
IT 9253	Embedded Laboratory	0	0	3	2
CS 9256	Web Technology Laboratory	0	0	3	2
CS 9257	Operating Systems Laboratory	0	0	3	2
TOTAL		18	1	9	25

Aim:

To facilitate the understanding of the principles and to cultivate the art of formulating physical problems in the language of mathematics.

Objectives:

- To introduce Fourier series analysis which is central to many applications in engineering apart from its use in solving boundary value problems
- To acquaint the student with Fourier transform techniques used in wide variety of situations in which the functions used are not periodic
- To introduce the effective mathematical tools for the solutions of partial differential equations that model physical processes
- To develop Z- transform techniques which will perform the same task for discrete time systems as Laplace Transform, a valuable aid in analysis of continuous time systems

1. FOURIER SERIES**9+3**

Dirichlet's conditions – General Fourier series – Odd and even functions – Half-range Sine and Cosine series – Complex form of Fourier series – Parseval's identity – Harmonic Analysis.

2. FOURIER TRANSFORM**9+3**

Fourier integral theorem – Fourier transform pair-Sine and Cosine transforms – Properties – Transform of elementary functions – Convolution theorem – Parseval's identity.

3. PARTIAL DIFFERENTIAL EQUATIONS**9+3**

Formation – Solutions of first order equations – Standard types and Equations reducible to standard types – Singular solutions – Lagrange's Linear equation – Integral surface passing through a given curve – Solution of linear equations of higher order with constant coefficients.

4. APPLICATIONS OF PARTIAL DIFFERENTIAL EQUATIONS**9+3**

Method of separation of Variables – Solutions of one dimensional wave equation and one-dimensional heat equation – Steady state solution of two-dimensional heat equation – Fourier series solutions in Cartesian coordinates.

5. Z – TRANSFORM AND DIFFERENCE EQUATIONS**9+3**

Z-transform – Elementary properties – Inverse Z-transform – Convolution theorem – Initial and Final value theorems – Formation of difference equation – Solution of difference equation using Z-transform.

L: 45, T: 15, Total : 60**TEXT BOOKS**

Grewal, B.S. "Higher Engineering Mathematics", Khanna Publications (2007)

REFERENCES

- 1) Glyn James, "Advanced Modern Engineering Mathematics, Pearson Education (2007)
- 2) Ramana, B.V. "Higher Engineering Mathematics" Tata McGraw Hill (2007).
- 3) Bali, N.P. and Manish Goyal, "A Text Book of Engineering 7th Edition (2007) Lakshmi Publications (P) Limited, New Delhi.

EC 9212 COMMUNICATION TECHNIQUES 3 0 0 3

1. ANALOG MODULATION 9

Amplitude Modulation – AM, DSBSC, SSBSC, VSB – PSD, modulators and demodulators – Angle modulation – PM and FM – PSD, modulators and demodulators – Superheterodyne receivers

2. DIGITISATION 9

Low pass sampling theorem – Quantisation - PAM – Line coding - PCM, DPCM, DM, ADPCM and ADM, Channel Vocoder,– Time Division Multiplexing, frequency Division Multiplexing

3. DIGITAL MODULATION AND TRANSMISSION 9

Phase shift keying – BPSK, DPSK, QPSK - Principles of M-ary signaling M-ary PSK & QAM – Comparison, ISI – Pulse shaping – Duo binary encoding - Cosine filters – Eye pattern, equalizers

4. INFORMATION THEORY AND CODING 9

Measure of information – Entropy – Source coding theorem - Shannon-Fano coding, Huffman Coding, LZ Coding– Channel capacity – Shannon-Hartley law – Shannon's limit- Error control Codes – Cyclic codes, Syndrome calculation – Convolutional Coding, Sequential and Viterbi decoding

5. SPREAD SPECTRUM AND MULTIPLE ACCESS 9

PN sequences – properties – m-sequence –DSSS –Processing gain, Jamming – FHSS –Synchronisation and tracking - Multiple Access – FDMA, TDMA, CDMA

TEXT BOOK:

1. H Taub, D L Schilling, G Saha, "Principles of Communication Systems" 3/e, TMH 2007
2. S. Haykin "Digital Communications" John Wiley 2005

REFERENCES:

1. B.P.Lathi, "Modern Digital and Analog Communication Systems", 3/e, Oxford University Press,2007
2. H P Hsu, Schaum Outline Series - "Analog and Digital Communications" TMH 2006
3. B.Sklar, "Digital Communications Fundamentals and Applications" 2/e Pearson Education 2007

Aim:

To provide a strong foundation in database technology and an introduction to the current trends in this field.

OBJECTIVES:

- To learn the fundamentals of data models and to conceptualize and depict a database system using ER diagram.
- To make a study of SQL and relational database design.
- To understand the internal storage structures using different file and indexing techniques which will help in physical DB design.
- To know the fundamental concepts of transaction processing- concurrency control techniques and recovery procedure.
- To have an introductory knowledge about the Storage and Query processing techniques

1.INTRODUCTION**9**

Purpose of Database System — Views of data – Data Models – Database Languages — Database System Architecture – Database users and Administrator – Entity–Relationship model – E-R Diagrams -- Introduction to relational databases

2.RELATIONAL MODEL**9**

The relational Model – The catalog- Types– Keys - Relational Algebra – Domain Relational Calculus – Tuple Relational Calculus - Fundamental operations – Additional Operations- SQL fundamentals - Integrity – Triggers - Security – Advanced SQL features –Embedded SQL– Dynamic SQL- Missing Information– Views – Introduction to Distributed Databases and Client/Server Databases

3.DATABASE DESIGN**9**

Functional Dependencies – Non-loss Decomposition – Functional Dependencies – First, Second, Third Normal Forms, Dependency Preservation – Boyce/Codd Normal Form- Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form

4.TRANSACTIONS**9**

Transaction Concepts - Transaction Recovery – ACID Properties – System Recovery – Media Recovery – Two Phase Commit - Save Points – SQL Facilities for recovery – Concurrency – Need for Concurrency – Locking Protocols – Two Phase Locking – Intent Locking – Deadlock- Serializability – Recovery Isolation Levels – SQL Facilities for Concurrency

5.IMPLEMENTATION TECHNIQUES**9**

Overview of Physical Storage Media – Magnetic Disks – RAID – Tertiary storage – File Organization – Organization of Records in Files – Indexing and Hashing –Ordered Indices – B+ tree Index Files – B tree Index Files – Static Hashing – Dynamic Hashing – Query Processing Overview – Catalog Information for Cost Estimation – Selection Operation – Sorting – Join Operation – Database Tuning.

TEXT BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, "Database System Concepts", Fifth Edition, Tata McGraw Hill, 2006 (Unit I and Unit-V) .
2. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.(Unit II, III and IV)

REFERENCES:

1. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Fourth Edition , Pearson / Addison wesley, 2007.
2. Raghu Ramakrishnan, "Database Management Systems", Third Edition, McGraw Hill, 2003.
3. S.K.Singh, "Database Systems Concepts, Design and Applications", First Edition, Pearson Education, 2006.

IF 9201 COMPUTER ORGANIZATION

3 1 0 4

Aim : To understand the basics of digital design, the design of various components of the computer system and its organization.

Objectives :

- To understand the fundamentals of Boolean logic and functions.
- To design and realize these functions with basic gates, and other components using combinational and sequential logic.
- To understand the design and organization of a von-neumann computer system.
- To comprehend the importance of the hardware-software interface.

1. Digital Fundamentals 9 +3

Number systems and conversions – Boolean algebra and simplification – Minimization Of Boolean functions – Karnaugh map – Quine McCluskey tabulation method – Logic gates – NAND – NOR implementation.

2. Combinational and Sequential Circuits 9+3

Design of combinational circuits – Adder / Subtractor – Encoder – Decoder – Mux / Demux – Comparators – Flip Flops – Triggering – Master – Slave Flip Flop – State diagrams and minimization – Counters – Registers.

3. Basic Structure of Computers 9+3

Functional units – Basic operational concepts – Bus structures – Performance and metrics – Instructions and instruction sequencing – Hardware – Software Interface – Instruction set architecture – Addressing modes – RISC – CISC. ALU design – Fixed point and floating point operations.

4. Processor Design 9 +3

Fundamental concepts – Execution of a complete instruction – Hardwired control – Micro programmed control. Pipelining – Basic concepts – Data hazards – Instruction hazards – Influence on instruction sets – Data path and control considerations – Performance considerations

5. Memory and I/O Systems 9 +3

Memory Technology – Memory hierarchy – Cache Memory – Design Methods – Virtual Memory – Input/Output System – Programmed I/O – DMA and Interrupts – Functions of I/O devices and interfaces.

TOTAL = 45+15

TEXT BOOKS:

1. Morris Mano, "Digital Design", Third Edition, Pearson Education, 2002.
2. Carl Hamacher, Zvonko Vranesic And Safwat Zaky, "Computer Organization", Fifth Edition, Tata McGraw Hill, 2002.

REFERENCES:

1. Charles H. Roth, Jr., "Fundamentals of Logic Design", Fifth Edition, Jaico Publishing House, 2003.
2. William Stallings, "Computer Organization and Architecture – Designing for Performance", Sixth Edition, Pearson Education, 2003.
3. David A. Patterson And John L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface", Third Edition, Elsevier, 2005.
4. John P. Hayes, "Computer Architecture and Organization", Third Edition, Tata McGraw Hill, 1998.

CS 9203 PROGRAMMING AND DATA STRUCTURES II 3 0 0 3

Aim:

The aim is to introduce the concepts Object Oriented Programming and analysis the implementation of Advanced Data Structures using Object Oriented Programming Language.

Objectives:

- To introduce the concepts of Object Oriented Programming language.
- To introduce the concepts of Templates and Error Handling.
- To introduce the concepts of Advanced Data Structures.

Unit I 9

Introduction – Learning C++ - Design of C++ - History and Use – Programming Paradigms – Standard Library – Types and Declaration – Pointers, Arrays, Structures – Expressions and Statements – Functions – Namespaces and Exceptions – Source

Unit II **9**

Type Conversion Operators – Friends – Large Objects – Essential Operators – Subscripting – Function Call – Dereferencing – Increment and Decrement – String Class – Derived Classes – Abstract Classes – Design of Class Hierarchies

Unit III **9**

Templates – Function Templates – Error Handling – Grouping of Exceptions – Catching Exceptions – Resource Management – Multiple Inheritance – Access Control – Run Time Type Information

Unit IV **9**

OO Perspective of List, Stack, Queue, and Search Tree ADTs – AVL Trees – Red Black Trees – Splay Trees – B-trees – Priority Queues (Heaps)

Unit V **9**

Disjoint Set ADT – Graph Algorithms – Topological Sort – Shortest-Path Algorithm – Network Flow Problems – Minimum Spanning Tree – Applications of Depth-First Search

Total: 45

TEXT BOOKS:

1. Bjarne Stroustrup, "The C++ Programming Language", 3rd ed., Pearson Education, 2007. (Units 1,2,3)
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 2nd ed., Pearson Education, 2005. (Units 4,5)

REFERENCE BOOKS:

1. Ira Pohl, "Object-Oriented Programming using C++", 2nd ed., Pearson Education, 1997.
2. Goodrich, Michael T., Roberto Tamassia, David Mount. Data Structures and Algorithms in C++. 7th ed, Wiley. 2004.

CS 9201 DESIGN AND ANALYSIS OF ALGORITHMS 3 0 0 3

Aim:

The aim is to introduce the basics of algorithm design paradigms and analysis to enable designing of efficient algorithms.

Objectives:

- To introduce the basic concepts of algorithm analysis
- To introduce the design paradigms for algorithm design
- To introduce the basic complexity theory.

UNIT I **9**

The Role of Algorithms in Computing-Getting Started-Growth of Functions – Recurrences-The Substitution Method- The Recurrence Tree Method-The Master Method -Probabilistic Analysis and Randomized Algorithms-The Hiring Problem-Random Variables-Randomized Algorithms.

UNIT II

9

Quicksort-Description-Performance-Randomized version-Analysis.Sorting in linear time-Lower bounds for sorting-Counting sort-Medians and order statistics-Minimum and maximum-Selection in expected linear time- Selection in worst-case linear time-Dynamic Programming – Matrix chain multiplication –Elements of Dynamic programming- Longest common sequences.

UNIT III

9

Greedy Algorithms-Activity selection problem-Elements of Greedy Strategy-Huffman code.Matrix Operations-Properties of matrices-Strassen's algorithm-Solving systems of linear equations-Inverting matrices.

UNIT IV

9

Linear Programming-Standard and slack forms-Formulating problems-Simplex algorithm-Duality-Initial basic feasible solution - String Matching-Naive string matching algorithm-Knuth-Morris-Pratt algorithm.

UNIT V

9

NP-completeness-Polynomial time-Polynomial-time verification-NP-completeness and reducibility-NP-completeness proofs - NP-completeness problems. Approximation Algorithms-The vertex-cover problem-The traveling-salesman problem.

TOTAL : 45

TEXT BOOKS

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein, "Introduction to Algorithms", Second Edition, Prentice Hall of India, 2007.

REFERENCE BOOKS:

1. Jon Kleinberg, Eva Tardos, "Algorithm Design", Pearson Education, 2006.
2. Michael T. Goodrich, Toberto Tamassisa, " Algorithm Design: Foundations, Analysis and Internet Examples", Wiley Student Edition, 2007.
3. Anany Levitin, "Introduction to Design and Analysis of Algorithms", Pearson Education, 2003.

CS 9205

DBMS LAB

0 0 3 2

Experiments in the following topics:

1. Data Definition, Manipulation of base tables and views
2. High level programming language extensions.
3. Front end tools

4. Forms
5. Triggers
6. Menu Design
7. Importing/ Exporting Data
8. Reports.
9. Database Design and implementation (Mini Project).

CS 9206 Programming and Data Structures Lab II

0 0 3 2

Experiments in the following:

1. Data abstraction, Implementation of any one of the following List, Stack, Queue ADTs, using Header files, Separate compilation of implementation and application. Search ADT, Binary Search Tree., Header files, Separate compilation.
2. Use of Standard Template Library: Strings, Containers
3. Use of STL: Iterators
4. Operator Overloading
5. Templates,
6. Exception handling, Class Hierarchies
7. AVL Tree
8. Splay Tree
9. B Tree
10. Graph algorithms

CS 9207 ALGORITHMS LAB

0 0 3 2

1. Implementing simple recursive programs like Towers of Hanoi ,Generating Permutations.
2. Implementation of simple sort algorithms.
3. Implementation of randomized quicksort algorithms.
4. Implementation of algorithms like merge sort using Divide and Conquer approach.
5. Implementation of Huffman code using Greedy Approach.
6. Implementation of Dynamic Programming – Floyd's Algorithm.
7. Solving Simplex Method.
8. Implement String matching algorithms.

9. Implementation of programs Study of Benchmarking and analysis of algorithms for simple programs.
10. Study of Algorithms Tools like LEDA (Library of Efficient Data types and Algorithms).

MA 9265 DISCRETE MATHEMATICS

3 1 0 4

AIM:

To extend student's Logical and Mathematical maturity and ability to deal with abstraction and to introduce most of the basic terminologies used in computer science courses and application of ideas to solve practical problems.

OBJECTIVES:

At the end of the course, students would

- Have knowledge of the concepts needed to test the logic of a program.
- Have an understanding in identifying structures on many levels.
- Be aware of a class of functions which transform a finite set into another finite set which relates to input output functions in computer science.
- Be aware of the counting principles
- Be exposed to concepts and properties of algebraic structures such as semi groups, monoids and groups.

1. Logic and Proofs

9 + 3

Propositional Logic – Propositional equivalences-Predicates and quantifiers – Nested Quantifiers – Rules of inference-introduction to proofs – proof methods and strategy.

2. Combinatory

9 + 3

Mathematical induction – Strong induction and well ordering – The basics of counting - The pigeonhole principle – Permutations and combinations – Recurrence relations-Solving linear recurrence relations-generating functions – Inclusion and exclusion and applications.

3. Graphs

9 + 3

Graphs and graph models – Graph terminology and special types of graphs - Representing graphs and graph isomorphism – connectivity – Euler and Hamilton paths.

4. Algebraic Structures

9 + 3

Algebraic systems – Semi groups and monoids – Groups-Subgroups and homomorphisms – Cosets and Lagrange's theorem – Ring & Fields.

5. Lattices and Boolean algebra

9 + 3

Partial ordering – Posets – Lattices as Posets – Properties of lattices-Lattices as algebraic systems – Sub lattices – direct product and Homomorphism – Some special lattices – Boolean algebra

L: 45, T: 15, Total : 60

TEXT BOOKS

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", 6th Edition, Special Indian edition, Tata McGraw – Hill Pub. Co. Ltd., New Delhi, (2007).
2. Trembly J.P. and Manohar R, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw–Hill Pub. Co. Ltd, New Delhi, 30th Re-print (2007).

REFERENCES

1. Ralph. P. Grimaldi, "Discrete and Combinatorial Mathematics: An Applied Introduction", Fourth Edition, Pearson Education Asia, Delhi, (2002).
2. Thomas Koshy, "Discrete Mathematics with Applications", Elsevier Publications, (2006).
3. Seymour Lipschutz and Mark Lipson, "Discrete Mathematics", Schaum's Outlines, Tata McGraw – Hill Pub. Co. Ltd., New Delhi, 2007, Second edition, Fifth reprint, (2007).

IT 9251 FORMAL LANGUAGES AND AUTOMATA 3 0 0 3

Aim:

To get fundamental idea on Automata and Languages

Objectives:

- A study of the formal relationships between machines, languages and grammar.

1. Automata 9

Introduction to formal proof – Additional forms of proof – Inductive proofs – Finite Automata (FA) – Deterministic Finite Automata (DFA) – Non-deterministic Finite Automata (NFA) – Finite Automata with Epsilon transitions.

2. Regular Expressions and Languages 9

Regular Expression – FA and Regular Expressions – Proving languages not to be regular – Closure properties of regular languages – Equivalence and minimization of Automata.

3. Context-free Grammars and Languages 9

Context-Free Grammar (CFG) – Parse Trees – Ambiguity in grammars and languages – Definition of the Pushdown automata – Languages of a Pushdown Automata – Equivalence of Pushdown automata and CFG – Deterministic Pushdown Automata.

4. Properties of Context-free Languages 9

Normal forms for CFG – Pumping Lemma for CFL – Closure Properties of CFL – Turing Machines – Programming Techniques for TM.

5. Deterministic Parsing 9

Parsing : The graph of a grammar – A top-down parser – Reductions and Bottom-up Parsing – A Bottom-up Parser – LL(k) Grammars : Lookahead in CFG – FIRST, FOLLOW, Lookahead sets – Strong LL(k) Grammars – Construction of FIRST_k, FOLLOW_k sets – LR(k) Grammars : LR(0) contexts – An LR(0) Parser – LR(0) Machine.

TOTAL = 45

Text Books:

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", second Edition, Pearson Education, 2003.
2. Thomas A. Sudukamp, "An Introduction to the Theory of Computer Science, Languages and Machines". 3rd Edition, Pearson Education, 2007.

References:

1. H.R. Lewis and C.H. Papadimitriou, "Elements of the theory of Computation", Second Edition, Pearson Education, 2003.
2. J. Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata Mc Graw Hill, 2003.
3. Micheal Sipser, "Introduction of the Theory and Computation", Thomson Brokecole, 1997.

IT 9252 EMBEDDED SYSTEMS

3 0 0 3

Aim : To understand hardware and the software aspects of embedded systems.

Objectives :

- To understand the architecture of embedded processors, microcontrollers, and peripheral devices.
- To appreciate the nuances of programming micro-controllers in assembly for embedded systems.
- To understand the challenges in developing operating systems for embedded systems.
- To learn about programming these systems in high-level languages such as C.

1. Embedded Computing

9

Challenges of Embedded Systems – Embedded system design process. Embedded processors – 8051 Microcontroller, ARM processor – Architecture, Instruction sets and programming.

2. Memory and Input / Output Management

9

Programming Input and Output – Memory system mechanisms – Memory and I/O devices and interfacing – Interrupt handling.

3. Processes and Operating Systems

9

Multiple tasks and processes – Context switching – Scheduling policies – Interprocess communication mechanisms – Performance issues.

4. Embedded C Programming

9

Programming embedded systems in C – C-looping structures – Register allocation – Function calls – Pointer aliasing – structure arrangement – bit fields – unaligned

5. Embedded System Development 9

Meeting real time constraints – Multi-state systems and function sequences.
Embedded software development tools – Emulators and debuggers. Design issues – Design methodologies – Case studies – Complete design of example embedded systems.

TOTAL = 45

TEXT BOOKS

1. Wayne Wolf, "Computers as Components:Principles of Embedded Computer System Design", Elsevier, 2006.
2. Muhammed Ali Mazidi, Janice Gillispie Mazidi and Rolin D. McKinlay, "The 8051 Microcontroller and Embedded Systems", Pearson Education, Second edition, 2007 (unit 1)
3. .Andrew N Sloss, D. Symes, C. Wright, " Arm system developers guide", Morgan Kauffman/ Elsevier, 2006. (unit 4)

REFERENCES

1. Michael J. Pont, "Embedded C", Pearson Education , 2007.
2. Steve Heath, "Embedded System Design" , Elsevier, 2005.

CS 9253 WEB TECHNOLOGY 3 0 0 3

Aim:

To provide an introduction to Java and basic Web concepts and enable the student to create simple Web based applications.

Objectives:

- To introduce the features of object oriented programming languages using Java
- To design and create user interfaces using Java frames and applets
- To have a basic idea about network programming using Java
- To create simple Web pages and provide client side validation
- To create dynamic web pages using server side scripting

Unit I 9

Java fundamentals – Class, Object – Inheritance – Polymorphism – Packages – Interfaces – Exception handling

Unit II 9

I/O – AWT – Event handling – Introduction to Threads - Basics of Networking –TCP and UDP sockets – Connecting to the Web

Unit III 9

Applets – JDBC – Swings – Remote Method Invocation

Unit IV 9

Unit V

9

Server side scripting – JSP – Servlets – Session management – Cookies

Total : 45

TEXTBOOK:

1. Deitel and Deitel, “Java – How to program”, 3rd ed., Pearson Education, 2001.
2. Robert W. Sebesta, “Programming the World Wide Web”, 3rd ed., Pearson Education, 2006. (Units 4,5)

REFERENCE BOOKS:

1. Herbert Schildt, “Java – The Complete Reference”, 7th ed., Tata McGraw Hill, 2007.
2. Chris Bates, “Web Programming”, 3rd ed., Wiley, 2006.
3. Black Book, “Java 6 Programming”, Dreamtech Press, 2007.
4. Deitel, “Java How to Program”, Pearson Education, 2003.
5. W Clay Richardson, et al, “Professional Java JDK 6 Edition”, Wrox, 2007.

CS 9252 OPERATING SYSTEMS

3 0 0 3

Aim: The course introduces the students to the basic principles of operating systems.

Objectives:

- To be aware of the evolution of operating systems
- To learn what processes are, how processes communicate, how process synchronization is done and how to manage processes
- To have an understanding of the main memory and secondary memory management techniques.
- To understand the I/O Subsystem
- To have an exposure to Linux and Windows 2000 operating systems

1. Operating Systems Overview

9

Operating system – Types of Computer Systems – Computer-system operation – I/O structure – Hardware Protection – System components – System calls – System programs – System structure – Process concept – Process scheduling – Operations on processes – Cooperating processes – Interprocess communication – Communication in client-server systems – Multithreading models – Threading issues – Pthreads.

2. Process Management

10

Scheduling criteria – Scheduling algorithms – Multiple-processor scheduling – Real time scheduling – Algorithm Evaluation – Process Scheduling Models - The critical-section problem – Synchronization hardware – Semaphores – Classic problems of synchronization – Critical regions – Monitors – System model – Deadlock characterization – Methods for handling deadlocks – Recovery from deadlock

3. Storage Management

9

Memory Management – Swapping – Contiguous memory allocation – Paging – Segmentation – Segmentation with paging. Virtual Memory: Background – Demand paging – Process creation – Page replacement – Allocation of frames – Thrashing.

4. I/O Systems 9

File concept – Access methods – Directory structure – File-system mounting – Protection – Directory implementation – Allocation methods – Free-space management – Disk scheduling – Disk management – Swap-space management.

5. Case Study 8

The Linux System – History – Design Principles – Kernel Modules – Process Management – Scheduling – Memory management – File systems – Input and Output – Inter-process Communication – Network Structure – Security – Windows 2000 – History – Design Principles – System Components – Environmental subsystems – File system – Networking.

TOTAL = 45

TEXT BOOKS:

1. Silberschatz, Galvin and Gagne, “Operating System Concepts”, Sixth Edition, John Wiley & Sons Inc 2003.

REFERENCES:

1. Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, Addison Wesley, 2001.
2. Gary Nutt, “Operating Systems”, Second Edition, Addison Wesley, 2003.
3. H M Deital, P J Deital and D R Choffnes, “Operating Systems”, Pearson Education, 2004.

CS 9254 SOFTWARE ENGINEERING 3 0 0 3

Aim:

The course is intended to give Software Engineering principles in classical sense.

Objectives:

- To be aware of a member of generic models to structure the software development process.
- To understand fundamental concepts of requirements engineering and requirements specification.
- To understand different notion of complexity at both the module and system level
- To be aware of some widely known design methods.
- To understand the role and contents of testing activities in different life cycle phases.

UNIT I 9

The Evolving role of Software – Software – The changing Nature of Software – Legacy software —A generic view of process– A layered Technology – A Process

Framework – The Capability Maturity Model Integration (CMMI) – Process Assessment – Personal and Team Process Models. Product and Process. Process Models – The Waterfall Model – Incremental Process Models – Incremental Model – The RAD Model – Evolutionary Process Models – Prototyping – The Spiral Model – The Concurrent Development Model – Specialized Process Models – the Unified Process.

UNIT II

9

Software Engineering Practice – communication Practice – Planning practice Modeling practice– Construction Practice –Deployment. Requirements Engineering - Requirements Engineering tasks – Initiating the requirements Engineering Process- Eliciting Requirements – Developing Use cases – Building the Analysis Models – Elements of the Analysis Model – Analysis pattern – Negotiating Requirements – Validating Requirements.

UNIT III

9

Requirements Analysis – Analysis Modeling approaches – data modeling concepts – Object oriented Analysis – Scenario based modeling – Flow oriented Modeling – Class based modeling – creating a behaviour model.

UNIT IV

9

Design Engineering – Design process -Design Quality-Design model-User interface Design – Testing strategies- strategies Issues for conventional and object oriented software-validation testing –system testing –Art of debugging – Project management

UNIT V

9

Software evolution - Verification and Validation -Critical Systems Validation – Metrics for Process, Project and Product-Quality Management -Process Improvement –Risk Management- Configuration Management

TOTAL: 45

TEXT BOOKS:

1. Roger S.Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill International edition, Sixth edition, 2005.
2. Ian Sommerville, Software Engineering, 8th Edition, Pearson Education, 2008(UNIT V)

REFERENCES:

1. Stephan Schach, Software Engineering, Tata McGraw Hill, 2007.
2. Pfleeger and Lawrence Software Engineering: Theory and Practice, Pearson Education, second edition, 2001

Aim : To get a feel of programming and developing an embedded system

Objectives:

- To write programs in micro-controller assembly language.
 - To learn to interface various peripheral devices with microcontrollers.
 - To understand real-time aspects of embedded systems.
1. 8051 Microcontroller based experiments- assembly language programs -2 experiments
 2. 8051 Microcontroller based experiments- Control applications
 3. 8051 Microcontroller based I/O interfacing - 3 experiments
 4. Testing RTOS Environment and System Programming (embedded C)
 - KEIL software example programs – 3 experiments
 5. Real time systems program – 2 experiments

CS 9256

WEB TECHNOLOGY LABORATORY

0 0 3 2

Aim:

To enable the students to program in Java and to create simple Web based applications.

Objectives:

To write simple programs using Java
To design and create user interfaces using Java frames and applets
To write I/O and network related programs using Java
To create simple Web pages and provide client side validation
To create dynamic web pages using server side scripting

Experiments in the following:

1. Java Fundamentals, Classes, Objects
2. Inheritance, Polymorphism
3. Interfaces, Exception handling
4. I/O, AWT
5. Socket Programming
6. Applets, Swings
7. Database connectivity
8. RMI
9. XML, Style sheet, Parser
10. Client side scripting
11. JSP, Servlets
12. Session Management

Total : 45

1. Basic UNIX commands.
2. Shell Programming.
3. Grep, sed, awk.
4. File system related system calls.
5. Process management – Fork, Exec.
6. Message queues.
7. Pipe, FIFO's.
8. Signals.
9. Shared memory.
10. Semaphores.

TOTAL = 45