COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY

B. Tech. (Computer Engineering): Third Year
Effective from A. Y. 2017-18

INDEX

Item **Page No.**

List of Abbreviations

Sr. No.	Abbreviation	Stands for:
1	DEC	Departmental Elective Course
2	PSC	Professional Science Course
3	PCC	Program Core Course
4	LC	Laboratory Course
5	HSSC	Humanities and Social Science Course
6	MLC	Mandatory Learning Course
8	LLC	Liberal Learning Course
9	BSC	Basic Science Course
10	SBC	Skill Based Course

Program Education Objectives (PEOs)

I. To create graduates with sufficient capabilities in computer engineering who can become researchers, entrepreneurs and software professionals to satisfy the needs of the core industry, research, academia and society at large.

II. To provide opportunity to learn the latest trends in computer engineering and prepare for lifelong learning process.

III. To make the students aware of professional ethics of the software Industry and prepare them with basic soft skills essential for working in community and professional teams.

.

Program Outcomes (POs)

At the end of the program, the graduates will

1. Computer engineering knowledge: Apply the knowledge of mathematics, science, computer engineering fundamentals, and emerging fields of computer engineering to the solution of complex real life problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex computer engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and computer engineering sciences.

3. Design/development of solutions: Design solutions for complex computer engineering problems and design system components or processes that meet the specified needs considering public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern computer engineering and IT tools including FOSS tools.

6. Social responsibility: Apply reasoning informed by the contextual knowledge to assess social, health, safety, legal and cultural issues and the consequent responsibilities.

7. Environment and sustainability: Understand the impact of the professional computer engineering solutions in socio-environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Demonstrate knowledge and practice of engineering ethics.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary, multi-cultural settings.

10. Communication: Communicate effectively with engineering community and with society at large, demonstrating ability to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the computer engineering, finance and management principles.

12. Life-long learning: Recognize the need for, and ability to engage in independent and life-long learning.

CURRICULUM STRUCTURE OF T. Y. - B.TECH (Computer Engineering)

Effective from A. Y. 2017-2018

Sem V: : Effective AY 2017-18

Sr. No.	Course Type	Course Name	Teaching Scheme L	T	P	Credits
1	BSC	Probability and Statistics for Computing	2	1	0	3
2	MLC	Constitution of India	1	0	0	0
3	ILOE in Humanities /HSMC	Engineering Economics / Industrial & Corporate Practices / Applied Psychology	2	0	0	2
4	PCC	Computer Organization	3	0	0	3
5	PCC	Database Management Systems	3	0	0	3
6	PCC	Artificial Intelligence	3	0	0	3
7	PCC	Computer Networks	3	0	0	3
8	LC	Database Management Systems Laboratory	0	0	2	1
9	LC	Artificial Intelligence Laboratory	0	0	2	1
10	LC	Computer Networks Laboratory	0	0	2	1
11	LC	Software Engineering: Mini Project - Stage 1	0	1	2	2
		Total Academic Engagement and Credits	17	1	6	22

Sem VI: : Effective AY 2017-18

Sr. No.	Course Type	Course Name	Teaching Scheme L	T	P	Credits
1	ILOE in Humanities /HSMC	Engineering Economics / Industrial & Corporate Practices / Applied Psychology	2	0	0	2
2	MLC	Environmental Studies	1	0	0	0
3	SLC	Technical MOOC/Industry floated Course	3	0	0	3
4	SBC	Software Engineering: Mini project - Stage II	2	0	2	3
5	DEC	Department Elective-I	3	0	0	3
6	PCC	Operating Systems	3	0	0	3
7	PCC	Algorithms and Complexity	3	0	0	3
8	PCC	Data Science	3	1	0	4
9	LC	Operating Systems Laboratory	0	0	2	1
10	LC	Department Elective-I Laboratory	0	0	2	1
11	HSMC	Entrepreneurship Development	1	0	0	1
		Total Academic Engagement and Credits	20	1	6	24

(CT) Probability and Statistics for Computing

Teaching Scheme:
Lectures : 2 Hrs/week
Tutorial : 1 Hr/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Solve problems related to basic probability theory
2. Solve problems related to basic concepts and commonly used techniques of statistics
3. Model a given scenario using continuous and discrete distributions appropriately and estimate the required probability of a set of events
4. Apply theory of probability and statistics to solve problems in domains such as machine learning, data mining, computer networks etc.
5. Demonstrate the use of R language for data analysis.

Unit I: Basic Probability Theory: Probability axioms, conditional probability, independence of events, Bayes' rule [02 Hrs]

Unit II: Random Variables: Discrete and continuous random variables; Discrete Distributions such as Binomial, Poisson, Geometric etc.; Continuous Distributions such as Exponential, Normal etc.; Expectation: Moments; Central Limit theorem and its significance; Some sampling distributions like chi-square, t, F [10 Hrs]

Unit III: Introduction to 'R': Introductory R language fundamentals and basic syntax, major R data structures, Using R to perform data analysis, creating visualizations using R [02 Hrs]

Unit IV: Statistical Inference: Estimation - introduction, classical methods of estimation, single sample: estimating the mean and variance, two samples: estimating the difference between two means and ratio of two variances; Tests of hypotheses - introduction, testing a statistical hypothesis, tests on single sample and two samples concerning means and variances; ANOVA - One–way, Two–way with/without interactions [12 Hrs]

Unit V: Regression and Correlation: Simple linear regression model, Least square estimators, polynomial regression, Correlation [02 Hrs]

Unit VI: Introduction to Queuing Theory: Stochastic Processes, Markov Processes and Markov Chains, Birth-Death Process, Basic Queuing Theory (M/M/-/-) Type Queues [02 Hrs]

Text Books:

- Ronald E, Walpole, Sharon L. Myers, Keying Ye, "Probability and Statistics for Engineers and Scientists", Pearson, 9th edition, ISBN-13: 978-9332519084
- V. Sundarapandian, "Probability, Statistics and Queuing Theory", PHI, 1st edition, ISBN-13: 978-8120338449

Reference Books:

- Sheldon M. Ross, "Introduction to Probability and Statistics for Engineers and Scientists", Elsevier, 4th edition, ISBN-13: 978-8190935685
- Kishor Trivedi, "Probability and Statistics with Reliability, Queuing, and Computer Science Applications", John Wiley and Sons, New York, 2001, ISBN number 0-471-33341-7

Useful links/web resources:

- http://nptel.ac.in/courses/117103017/
- Introduction to R for Data Science on edX

(CT) Computer Organization

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Analyze Instruction set architecture, control signals in CPU, Hard wired control & Microprogrammed control units.
2. Apply Booth algorithm for multiplication, floating point number representation & Arithmetic.
3. Describe DRAM technology, cache memory, paging in virtual memory and Secondary Storage.
4. Explain multiprocessor system & bus arbitration, Instruction pipelining & RISC.

Unit I : CPU Architecture: instruction format, control signals in CPU, micro program control unit and hard wired control unit, ALU & sequencer, look ahead carry generator. **[6 Hrs]**

Unit II : Arithmetic: Integer Arithmetic-multiplication, Booth's Algorithm, division algorithm; Floating point number representation, and floating point arithmetic. **[6 Hrs]**

Unit III : Memory: Dynamic RAM organization, CACHE memory & it's mapping, cache coherence & MESI protocol, virtual memory, secondary storage, MBR and GPT hard disks, IDE, SCSI, RAID, CD, DVD,SSD, File system FAT, NTFS **[7 Hrs]**

Unit IV : System and memory map : closely coupled and loosely coupled multiprocessor systems, bus arbitration, co processor, Desktop key board, lower 1MB memory map & its video RAM, character generator ROM, Monochrome display adapter & color/graphics adapter **[8 Hrs]**

Unit V : Instruction Pipelining: Basic concepts and issues, Introduction to the basic features & architecture of RISC & CISC processors, super scalar processor. **[7 Hrs]**

Unit VI : Multiprocessor: Introduction to Multicores, Multiprocessors and Clusters. **[6 Hrs]**

Text Books:

- William Stallings, Computer Organization and Architecture, 9/E ISBN-10: 013293633X ISBN-13: 9780132936330©2013 Prentice Hall
- Carl Hamacher, Zvonko Vraesic and Safwat Zaky, Computer Organisation, ISBN 0-07-232086-9, MGH 5th edition.

Reference Books:

- Liu & Gibson, Microcomputer Systems, PHI, ISBN: 978-81-203-0409-3
- Douglas V. Hall, Microprocessors and interfacing, ISBN 0-07-025526-1,Tata McGraw-Hill
- D. Paterson, J. Hennesy, “Computer Organization and Design: The Hardware Software Interface”, 2nd Edition, Morgan Kauffman, 2000 ISBN 981 – 4033 – 588.

(CT) Database Management Systems

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Identify and describe various components of DBMS.
2. Construct Entity-Relationship Model for given applications and Relational Model for the same.
3. Design and write best possible or optimal (SQL) query statement for given statement.
4. Apply normalization to database design.
5. Improve efficiency of data retrieval using various storage systems and indexing.
6. Describe concurrency control protocol and solve analytical problems on serializability.

Unit I : Introduction: Basic Concepts; Database system application, purpose of database systems, view of data, database languages; Database architecture: components of DBMS and overall structure of DBMS; Various types of databases **[6 Hrs]**

Unit II : E-R and Relational Model: Database design; E-R model: modeling, entity, attributes, relationships, constraints, components of E-R model; Relational model: basic concepts, attributes and domains, concept of integrity and referential constraints, schema diagram. **[8 Hrs]**

Unit III : Relational Algebra and SQL: Relational algebra**:** fundamental relational algebra operations, additional relational algebra operations, extended relational algebra operations, null values, modification to database; SQL: basic structure and operations, aggregate functions, nested subqueries, complex queries, views. **[6 Hrs]**

Unit IV : Relational Database Design: Basic concept of normalization; Decomposition using functional dependencies. **[6 Hrs]**

Unit V : Indexing and Hashing: Basic of query processing; Indices: concepts, B+ trees and B tree index file; Static and dynamic hashing. **[6 Hrs]**

Unit VI : Transactions and Concurrency control: Transaction: basic concepts, states, concurrent execution, serializability, recoverability, isolation; Concurrency control: timestamps and locking protocols, validation based protocols, multiple granularity protocols, deadlock handling; Recovery. **[8 Hrs]**

Text Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database system concepts”, Fifth Edition, McGraw Hill International Edition, ISBN 978-0073523323.
- Raghu Ramkrishnan, Johannes Gehrke, “Database Management Systems”, Second Edition, McGraw Hill International Editions, ISBN 978-0072465631.

Reference Books:

- Rob Coronel, “Database systems : Design implementation and management”, Forth Edition, Thomson Learning Press, ISBN 978-1418835934.
- Ramez Elmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Third Edition, Pearson Education, 2003, ISBN 978-0321204486.
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", Second Edition, Pearson, 2008, ISBN 978-0131873254.

(CT) Database Management Systems Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Continuous evaluation: 50 Marks
Practical Exam: 50 Marks

Course Outcomes:

Shared with the theory course: "Database Management Systems"

Suggested List of Assignments:

1. Write simple SQL Queries on the given schema.
2. Write SQL queries using aggregates, grouping and ordering statements for given statements on given schema.
3. Write SQL queries for given schema using Nested Subqueries and SQL Updates
4. Write DDL and DML statements for given statements.
5. Create the schema and constraints on the given relations using given statements.
6. Demonstrate database connectivity through High Level Programming Language.
7. Select any real time problem for database implementation. Draw an ER diagram for the selected problem in hand. Normalise the database up to appropriate normal form.
8. Analyze indexing and query processing. (Goal: to show a query whose plan uses an index and another that not using any index and must do an expensive scan on the same relation, and show the difference in run times).

This list is a guideline. The instructor is expected to improve it continuously.

(CT) Artificial Intelligence

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate knowledge of basics of the theory and practice of Artificial Intelligence
2. Write programs to demonstrate Artificial Intelligence techniques
3. Apply knowledge representation techniques and problem solving strategies to common AI applications.

Unit I : Introduction: What is AI, History, AI problems, Production Systems, Problem characteristics, Intelligent Agents, Agent Architecture, AI Application (E-Commerce, & Medicine), AI Representation, Properties of internal representation, Future scope of AI , Issues in design of search algorithms. **[6 Hrs]**

Unit II : Heuristic search techniques: Heuristic search, Hill Climbing, Best first search, mean and end analysis, Constraint Satisfaction, A* and AO* Algorithm, Knowledge Representation: Basic concepts, Knowledge representation Paradigms, Propositional Logic, Inference Rules in Propositional Logic, Knowledge representation using Predicate logic, Predicate Calculus, Predicate and arguments, ISA hierarchy, Frame notation, Resolution, Natural Deduction **[6 Hrs]**

Unit III : Logic Programming: Introduction, Logic, Logic Programming, Forward and Backward reasoning, forward and Backward chaining rules. Knowledge representation using non monotonic logic: TMS (Truth maintenance system), statistical and probabilistic reasoning, fuzzy logic, structure knowledge representation, semantic net, Frames, Script, Conceptual dependency. **[6 Hrs]**

Unit IV : Learning: What is Learning, Types of Learning (Rote, Direct instruction Analogy, Induction, Deduction) Planning: Block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, Least commitment strategy. **[6 Hrs]**

Unit V : Advance AI Topics: Game playing: Min-max search procedure, Alpha beta cutoffs, waiting for Quiescence, Secondary search, Natural Language Processing: Introduction, Steps in NLP, Syntactic Processing, ATN, RTN, Semantic analysis, Discourse & Pragmatic Processing. Perception and Action: Perception, Action, Robot Architecture **[8 Hrs]**

Unit VI : Neural Networks and Expert systems: Introduction to neural networks and perception-qualitative Analysis, Neural net architecture and applications, Utilization and

functionality, architecture of expert system, knowledge representation, two case studies on expert systems **[8 Hrs]**

Text Books:

- Elaine Rich and Kerin Knight, Artificial Intelligence, 2nd Edition, McGraw Hill, ISBN-13: 978-0070522633
- Eugene, Charniak, Drew Mcdermott, Introduction to artificial intelligence, Addison-Wesley, ISBN-13: 978-0201119459

Reference Books:

- Stuart Russell and Peter Norvig, Artificial Intelligence : A Modern Approach, Prentice Hall, 2nd Edition
- Ivan Bratko, Prolog Programming For Artificial Intelligence , 2nd Edition, Addison Wesley, 1990
- Herbert A. Simon, The Sciences of the Artificial, MIT Press, 3rd Edition, 1998
- Tim Jones, Artificial Intelligence Application Programming, M. Dreamtech Publication.
- George F Luger, Artificial Intelligence : Structures and Strategies for Complex Problem Solving, Pearson Edu., 4th Edition.
- Rajendra Akerkar, Introduction to Artificial Intelligence, PHI Publication

(CT) Artificial Intelligence Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Continuous evaluation: 50 Marks
End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Implement programs demonstrating Artificial Intelligence algorithms.
2. Implement a mini expert system.

List of Assignments:

1. Implement A* algorithm .
2. Implement AO* algorithm .
3. Implementation of Unification Algorithm.
4. Implementation of Truth maintenance system using prolog.
5. Implementation of Min/MAX search procedure for game Playing .
6. Parsing Method Implementation using Prolog.
7. Development of mini expert system using Prolog / Expert System Shell " Vidwan"

This list is a guideline. The instructor is expected to improve it continuously.

(CT) Computer Networks

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Analytically compare the state-of-the-art protocols and architectures in computer networks
2. Solve subnetting problems and analyze various routing mechanisms
3. Analyse network hardware components at the appropriate layers
4. Explain the concepts of datagram and have a hands-on experience of internet socket programming
5. Inspect the networking applications used in everyday tasks such as reading email or surfing the web and analyse its architecture
6. Design & implement protocols and network stacks for example scenarios

Unit I : Network Layer : Network layer services, IPv4, Problems with IPv4, strategies to bridge the limitations (IP subnetting, CIDR, DHCP, NAT), Network design with CIDR, IPv6 **[8 Hrs]**

Unit II: Network Layer Protocols: Routing algorithms: Unicast protocols: RIP, OSPF, BGP and multicast routing protocols, ICMP, IGMP, DHCP **[8 Hrs]**

Unit III: Transport Layer: Protocols Services, Transport layer protocols, UDP, TCP: State Transition diagram, flow control, error control, TCP Timers **[6 Hrs]**

Unit IV: Congestion control and Quality of Service: Queueing and QoS , TCP Congestion control, Congestion Avoidance Mechanisms **[6 Hrs]**

Unit V: Advanced Internetworking: Multicast Routing, Multiprotocol Label Switching (MPLS), Mobile IP, VoIP **[6 Hrs]**

Unit VI Applications: Traditional Applications (WWW, HTTP, FTP, Email, Telnet, SSH, DNS), Peer-to-Peer Networks, Socket programming, Security - firewall, DoS. **[6 Hrs]**

Text Books:

- Behrouz A. Forouzan, Firouz Mosharraf, Computer Networks: A Top-Down Approach, Tata McGraw-Hill Education Pvt. Ltd, ISBN 10: 1259001563 / ISBN 13: 9781259001567
- Peter Loshin,IPv6 Theory, Protocol, and Practice, Elsevier, ISBN: 9781558608108

References:

- Larry Peterson Bruce Davie, Computer Networks A Systems Approach, Elsevier, ISBN: 9780123850591
- Kevin R. Fall, W. Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols, Pearson, ISBN-13: 978-0321336316/ISBN-10: 0321336313
- Behrouz Forouzan , Data Communications and Networking , Tata McGraw-Hill, ISBN-13: 978-0073250328/ISBN-10: 0073250325
- William Stallings, “Data and computer Communication”, Pearson Education, ISBN-81-297-0206-1
- A S Tanenbaum, “Computer Networks”, Pearson Education, ISBN 9788177581652
- Alberto Leon Garcia and Indra Widjaja, “Communication Networks, Fundamental Concepts and Key Architectures”, Tata McGraw-Hill, ISBN-10: 007246352X
- J.F. Kurose and K. W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet” , Pearson, ISBN-13: 9780201976991

(CT) Computer Networks Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Term Work: 50 Marks
Oral/End Term Exam: 50 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the ability to use network analysis / monitoring tools to inspect the internal working of internetworking devices
2. Explain a programming language's internet socket programming interface and use it to build network applications
3. Implement tools for assisting subnet network calculations
4. Configure network emulation tools and use them to design virtual networks
5. Apply the knowledge of networking algorithms and implement various protocol simulations

List of Assignments:

1. Implement a subnet calculator whose basic functionality is to compute: subnet mask, classes, network ids, broadcast addresses, and number of subnets. Design a well-thought interface, which accepts the necessary inputs and displays the corresponding outputs.
2. Implement a client-server model using socket programming. You can choose your own language of choice (for e.g., C, C++, Java, Python) to implement a proof of concept client-server demo. Explain the socket programming APIs that you used in your program.
3. Implement RIP and OSPF routing protocol and demonstrate the working of the protocol as a simulation.
4. Implement TCP's flow-control and congestion control algorithms and demonstrate the working of the algorithms as a simulation.
5. Use Wireshark Interface for inspecting traffic at the DLL, network and transport layer and perform analysis to answer questions about the network traffic.
6. Design your own virtual network using the Mininet network emulator and interact with various hosts on this network.
7. Make a list of the network components and resources such as hubs, switches, routers, firewalls etc used in the COEP Institute Campus wide network. Perform an in-depth analysis of the network design and describe the reasoning for at least one of the design decisions taken as a part of the campus network.

This list is a guideline. The instructor is expected to improve it continuously.

Software Engineering (Mini Project) – Stage 1

Teaching Scheme:	**Examination Scheme:**
Laboratory : 2 Hrs/Week	Continuous evaluation: 70 Marks
Tutorial: 1 Hr / Week	End Semester Exam: 30 Marks

Course Outcomes:-

Students will be able to-

1. Demonstrate the use of tools and technologies used in software project development process.
2. Demonstrate the ability to communicate, solve technical problems, work in teams, and contribute to an ongoing software project.

Text Books/ Study Material/ Web Resources:

- "Debian New Maintainers' Guide", www.debian.org/doc/manuals/maint-guide
- Pro Git Book https://git-scm.com/book
- Autotools, GNU Manuals www.gnu.org/software/autoconf/
- GNU Gettext Manual https://www.gnu.org/software/gettext/manual/gettext.html
- Advanced Bash Scripting Guide, http://tldp.org/LDP/abs/html/

Suggested list of assignments:

- Write shell scripts for following tasks: convert a CSV file to VCF format, convert a youtube transcript to SRT format, find the top 10 size files created in last 20 days, move all duplicate files (except one) from a folder to a target location, etc.
- Write shell scripts or scripts in any language of your choice, to run conformance tests on a software of your choice.
- Create a git remote repository on any of the git hosting websites, using one of the software you have written so far. In a group of three or more people, carry out the following activities: reporting of bus, assigning of issues, fixing bugs, git branch and git pull requests.
- Localise and/or Internationalize any software and demonstrate your contributions. You may select any existing free software project for the same.
- Configure any of your existing C projects of atleast 500 lines using Autotools or Cmake or scons or any similar tool. You should write the required configuration files (like configure.in, Makefile.am files etc.) and also write a bootstrap program if needed.
- Package your software for Debian, Ubuntu, any Unix or other operating systems. For free software operating systems you should get your packaged software accepted by the respective communities.
- Fix bugs in any existing software, preferably a open source software by participating in the community development process.

This list is a guideline. The instructor is expected to improve it continuously.

Sem VI

(CT -) Software Engineering Mini Project Stage- II

Teaching Scheme:
Lectures : 2 Hrs/week
Laboratory: 2 Hrs/Week

Examination Scheme:
Quiz/Assignments: 30 Marks
Project: 70 Marks (Submission in Stages)

Course Outcomes

Students will be able to-

1. Describe fundamental concepts of system development lifecycle through SDLC.
2. Design user interface prototypes for real world scenarios using appropriate methods of analysis and design.
3. Devise procedure to assure the quality and maintainability of the product before and after deployment.
4. Develop skill to transfer acquired knowledge across a wide range of industrial and commercial domains and have a basis for further studies in software engineering or in computing related industries.
5. Analyze real world scenario and apply tools and techniques to produce application software solutions from informal and semiformal problem specifications.
6. Develop an ability to work in a team by communicating computing ideas effectively in writing a technical report.

Laboratory Mini Project Task

Students will carry out one of the following mini projects:

A) Work on an existing free software/open source project and contribute to it in terms of feature improvements or significant bug fixes. The project will be finalised in consultation with the laboratory instructor. The work can be carried out in teams of any size, however individual evaluation will be carried out on individual basis. The contributions should clearly bring out the following:
   1. Use of version control systems
   2. Contributions of each individual member of the team as seen in version control system
   3. Use of industry standard coding practices
   4. Participation in the software development process of the particular software
   5. Writing test cases and testing the software
   6. Deployment changes, Packaging if needed
   7. Acceptance of your contributions by the upstream community.

B) A full-fledged working system in the form of mini project will be implemented following the task list given below. Students in group of two will be working on mini project. After consultation with the course instructor and finalisation of the topic following deliverables are expected under mini-project. Task List for the same is as follows:
   1. Carry out state of art survey, selecting appropriate domain, problem identification,

statement formulation based on research problems or real-world problems, industry based problem etc.

2. Develop workflow graph and carry project estimation, calculation of efforts, project planning (schedule) using automated tools.
3. Gather requirements and Write the Software Requirement specification (SRS-IEEE specs) document for the project.
4. Draw different UML diagrams and System architecture for the proposed system. Use different open source tools for design.
5. Develop Test cases. Propose solution for wrong results in test cases by focusing on regression testing.
6. Write the constraints, advantages and disadvantages of your project over existing system.
7. Write the future scope of your project. Develop help manual for maintenance and usability.

Students will be required to submit a technical report written using LaTEX. The technical report will include description of the project/problem, design of the software, description of problems solved and solution design, result analysis of test cases and conclusions. Students will also be required to demonstrate and present their work in a viva-voce.

Syllabus

Unit 1: **Software Development process**: Software Engineering basics, Software Crisis and Myths, Software Process and development, Software life cycle and Models, Analysis and comparison of various models, agile process. [6 Hrs]

Unit 2: Requirement Engineering: Requirements Engineering, requirement engineering process, Introduction to Analysis model. [6 Hrs]

Unit 3: System Architecture and Design Overview: Architecture 4+1 view, architecture styles, Design process, quality concepts, design Model, Standardisation using UML. [6 Hrs]

Unit 4: Software Metrics: Introduction to Software Metrics, Size-oriented metrics and function point metrics. Effort and cost estimation techniques -LOC-based and Function-point based measures - The COCOMO model. [6 Hrs]

Unit 5: Testing: Validation and Verification activities, Testing Principles and strategies, Testing levels & types- White Box & Black Box Testing. [6 Hrs]

Text Books:

- Pressman R., "Software Engineering, A Practitioners Approach", 6th Edition, Tata McGraw Hill Publication, 2004, ISBN 007-124083-124083-7.
- G. Booch, J. Rumbaugh and I. Jacobson. The Unified Modeling Language User Guide, Addison Wesley, 1999.

Reference Books:

- Shari Pfleeger, “Software Engineering”, 2nd Edition. Pearson’s Education, 2001.
- Ian Sommerville, “Software Engineering”, 6th Edition, Addison-Wesley, 2000.
- Pankaj Jalote, “An Integrated Approach to Software Engineering”, Narosa publication house.
- Fred Brooks, “Mythical Manmonths”, www.cs.drexel.edu/~yfcai/CS451/.

Papers:

- Fred Brook, “No Silver Bullets”, IEEE Software 1987.
- Eric Raymond, “Cathedral and Bazaar “, www.tuxedo.org/~esr/writings.
- David Parnas, “On the Criteria To Be Used in Decomposing Systems into Modules”, Communications of the ACM, volume 15, #12, 1972.
- Grady Booch , IEEE software on architecture ,IEEE Computer Society.

(CT) Operating Systems

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Write programs to manipulate processes, files, and hardware resources using appropriate system calls.
2. Illustrate the design issues, solutions and complexity of operating system by compiling, modifying an OS kernel, tracing the sequence of activities on processor, data structures of a file system, race conditions, locking mechanisms and storage techniques.
3. Correlate the computer architecture features with operating system design issues.
4. Make design choices for an operating systems with given constraints

Unit I : Introduction and Operating Systems structures: Evolution of operating systems: Batch , timesharing, multiprogramming, multi tasking and distributed and real time. Operating system components, O.S. Services, System Calls, System Programs, System Structure, Virtual Machines, Special purpose operating systems, Open-source operating systems, Boot Procedure, Overview of the GNU/Linux system administration. **[4 Hrs]**

Unit II : Processes and CPU Scheduling: Process concept, interleaved I/O and CPU burst; Process states; Co-operating processes, Thread, Thread libraries, Multithreaded programming, Scheduling, Scheduling criterion, Scheduling algorithms, Multi processor scheduling, Real time scheduling, Interrupts and Interrupt handling. **[6 Hrs]**

Unit III : Process Synchronization: Critical section problem, Hardware support for mutual exclusion, Semaphores, Deadlock-principle, Deadlock detection, prevention and avoidance, Classical problems in concurrent programming: Producer-consumer, Reader-writer with and without bounded buffer. Design of locking primitives like spinlock, semaphore, read-write locks, recursive locks, etc. **[8 Hrs]**

Unit IV : **Inter process Communication:** Pipes, Shared memory mechanism, Streams, Asynchronous communication, Signals. Operating system interfaces for application programming using openMP and MPI, Client Server Computing, Remote procedure calls. **[4 Hrs]**

Unit V : Memory management: O.S. and hardware interaction, Swapping, Continuous memory management, paging, Segmentation, Virtual Memory Management, Demand Paging, Copy-on-write, Page replacement algorithms, Allocation of frames, Thrashing, Kernel memory management, SVR4 architecture, Unified buffer cache. **[8 Hrs]**

Unit VI : File Management and Storage Structures: File Organization, Concept of files and directories, System calls for file systems, Space allocation issues, Free space management, Data

structures like inode and super block, Virtual file system and related object oriented concepts, Disk layout, Ext2 disk layout, Formatting, Recovery, NFS, Efficiency and performance, Distributed file systems, Disk Structure, Disk Scheduling, RAID **[8 Hrs]**

Unit VII : Protection and Security

Goals of protection, Domain of protection, Access matrix, Implementation of access matrix, Revocation of access rights, Security problems, Authentication, Program threats, System threats, Threat monitoring. **[4 Hrs]**

Text Books:

- Abranhan Silberschatz, Peter B Galvin, Greg Gagne; Operating System Concepts, Wiley India Students Edition, 8th Edition, ISBN: 978-81-265-2051-0
- Andrew S. Tanenbaum; Modern Operating Systems; Prentice Hall of India Publication; 3rd Edition. ISBN: 978-81-203-3904-0

Reference Books:

- Milan Milenkovic; Operating Systems; Tata McGraw Hill; Second Edition. ISBN: 0-07-044700-4
- Maurice J. Bach; The Design of the Unix Opearating System; Prentice Hall of India; ISBN: 978-81-203-0516-8
- Uresh Vahalia; Unix Internals, The New Frontiers; Prentice Hall; ISBN: 0-13-101908-2

(CT) Algorithms and Complexity

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Determine different time complexities of a given algorithm
2. Demonstrate algorithms using various design techniques.
3. Develop algorithms using various design techniques for a given problem.
4. Formalize and abstract from a given computational task relevant computational problems, reduce problems and argue about complexity classes

Unit I: Introduction: Objectives of time and space analysis of algorithms; Order notations (O, Ω, θ notations); (Best average and worst case) time complexity of algorithms such as bubble sort, selection sort, insertion sort, heap sort etc.; Time complexity of recursive programs using recurrence relations. **[06 Hrs]**

Unit II: Design Techniques-I: Divide and Conquer: Quicksort, Mergesort, Strassen's matrix multiplication, finding convex hull; Greedy Algorithms: Knapsack problem, Job sequencing with deadlines, Optimal merge patterns, Single source shortest paths. **[06 Hrs]**

Unit III: Design Techniques–II: Dynamic Programming: All pairs shortest paths, 0-1 Knapsack, Traveling salesperson problem, Chained matrix multiplication, Longest common subsequence etc **[08 Hrs]**

Unit IV: Selected Algorithms from various areas: String Matching: The naïve string-matching algorithm, The Robin-Karp algorithm, The Knuth- Morris-Pratt algorithm; Polynomials and the FFT: Representation of polynomials, the DFT and FFT, efficient FFT implementations; Number - Theoretic algorithms: GCD algorithm, Chinese remainder theorem, Primality testing**.** **[08 Hrs]**

Unit V: Amortised Analysis: Aggregate analysis, accounting method, potential method, dynamic tables; Fibonacci heaps: measurable-heap operations, decreasing a key and deleting a node, bounding the maximum degree; Binomial heaps **[06 Hrs]**

Unit VI: Complexity Theory: Lower-bound arguments: Comparison Trees – sorting, Oracles and adversary arguments – merging, finding largest and second largest number in an array; NP-hard and NP-complete problems, proving NP-completeness using reduction technique (e.g. SAT, Independent Set, 3VC, Subset Sum, etc) **[06 Hrs]**

Text Books:

- Thomas Cormen, Charles Leiserson, Ronald Rivest and Cliford Stein, “Introduction to Algorithms”, PHI, 3rd edition, ISBN-13: 978-8120340077
- Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press, 2nd edition (2008), ISBN-13: 978-8173716126

Reference Books:

- Gilles Brassard and Paul Bratley, “Fundamentals of Algorithmics”, PHI, ISBN-13: 978-8120311312
- Jon Kleinberg and Éva Tardos, “Algorithm Design”, Pearson Education India, ISBN-13: 978-9332518643

(CT) Data Science

Teaching Scheme:
Lectures : 3 Hrs/week
Tutorials: 1 Hr/Week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Design data preprocessing model and demonstrate the working on every data type.
2. Design charts and demonstrate data analysis.
3. Apply different similarity measures, distance measures to find similarity or distances between data.
4. Demonstrate the handling of very large data using MapReduce.
5. Apply machine learning techniques to solve big data problem.

Unit I : Introduction: Introduction to Data Science, Examples, Data Sources, Challenges, Applications, Comparative Study of data science with databases, scientific computing, computational science, machine learning, Data Modeling, Statistical Data Modeling, Computational Data Modeling, Statistical limits on data- Bonferroni's principle. **[6 Hrs]**

Unit II : Data Preprocessing: Data types, Data preparation- data models, nosql data sources, data spaces, data cleaning and integration. Text data preprocessing- POS tagging, Bag of words, n-gram modeling **[6 Hrs]**

Unit III : Exploratory Data Analysis: Descriptive and inferential statistics, Chart types- Single var: Dot plot, Jitter plot, Error bar plot , Box-and-whisker plot, Histogram, Kernel density estimate, Cumulative distribution function, Two variable: Bar chart, Scatter plot, Line plot, Log-log plot, More than two variables: Stacked plots, Parallel coordinate plot, mean, variance, Hypothesis testing-T-test, CHI-squared and Fisher's test, ANOVA, K-S test, Permutation test, Bootstrap confidence intervals. **[6 Hrs]**

Unit IV : Similarity Measures, Distance Measures and Frequent Itemsets: Feature extraction - TF, IDF, TF-IDF, Hash functions, Similarity measuring techniques- Shingling, Min-hashing, Locality Sensitive hashing, Distance measures- Triangle Inequality, Euclidean Distance, Cosine Distance, Jaccard Distance, Edit Distance measures, Frequent Itemsets, the Market-Basket Model, Association Rules, A-Priori Algorithm, PCY (Park-Chen-Yu) Algorithm, Dimensionality reduction- UV decomposition, Singular-Value decomposition, CUR Decomposition. **[8 Hrs]**

Unit V : MapReduce and Search Engine Technologies: Distributed file system, physical organization of computer nodes, large-scale file system organization, MapReduce- map tasks, grouping by key, reduce tasks, combiners, MapReduce execution, Algorithm using MapReduce-

Matrix-Vector Multiplication by MapReduce, technology of Search Engines such as PageRank, link-spam detection, hubs-and-authorities. **[8 Hrs]**

Unit VI : Supervised-Un-Supervised-Reinforcement learning: Introduction, Supervised learning- Classification, Linear regression, Logistic Regression, Naive Bayes, Decision trees, Support Vector Machines, k-NN, Unsupervised learning: Clustering, k-means, Topic models, Reinforcement learning- Model quality, Over-fitting, Cross-validation, Precision, Recall, f-measure, Weighted Loss **[10 Hrs]**

Text Books:

- Cathy O'Neil and Rachel Schutt , "Doing Data Science", O'Reilly Media, October 2013 , Print ISBN:978-1-4493-5865-5| ISBN 10:1-4493-5865-9.
- Jure Leskovec, Anand Rajaraman, and Jeffery David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2 edition (13 November 2014), ISBN-10: 1107077230, ISBN-13: 978-1107077232.
- Tom Mitchell, "Machine Learning", McGraw-Hill, 1st Ed May 2013, ISBN-10: 1259096955| ISBN-13: 978-1259096952.

Reference Books:

- Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Pearson Education, First edition (2011), ISBN-10: 8131716724 , ISBN-13: 978-8131716724.
- Wes McKinney, "Python for Data Analysis", O'Reilly Media, October 2012, Print ISBN:978-1-4493-1979-3| ISBN 10:1-4493-1979-3.
- Garrett Grolemund," Hands- on Programming with R", O'Reilly Media (Kindle)

(CT) Operating Systems Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/week

Examination Scheme:
Continuous evaluation: 70 Marks
End Semester Exam: 30 Marks

Course Outcomes:

Shared with the theory course: "Operating Systems"

Suggested List of Assignments:

1. Create two virtual machines using virtual box software. One virtual machine will run a GNU/Linux of your choice on it. The other virtual machine will run any non-Linux operating system.
2. Write a minimal version of a shell. The shell should be able to a) execute a program without the complete path name b) handle pipes c) handle redirection d) run processes in background
3. Write a program which creates exactly 16 copies of itself by calling fork() only twice within a loop. The program should also print a tree of the pids.
4. Trace and explain completely the output of strace on running a "Hello World" C program.
5. Use debugfs tool to locate a file which was recently deleted on an ext2 file system. Write a program, on the lines of debugfs, to browse an ext2 file system and given the complete name name of a file, print it's inode.
6. Download linux kernel source code, compile it and reboot your system with the newly compiled kernel. Add a dummy system call to the Linux kernel. Write a conformance test to test your system call.
7. Read the GNU/Linux source code and show the code path which converts a call to read() system call to a file system specific read function call.
8. Implement a list type. Write a code using pthreads for concurrent insertions to the list and demonstrate the problem of race. Then rewrite the program to show how race conditions can be solved by using proper synchronization primitives.
9. Write a program which results in a guaranteed deadlock among it's threads.
10. Write a program using pthreads to demonstrate the producer consumer problem. Implement appropriate synchronization. Show the different results with and without synchronization.
11. Demonstrate the changing memory map of a process, by using the contents of the /proc file system, and creative use of malloc() function in the code of the process.
12. Write a program to demonstrate the usage of signals – show how processes can wait for each other, kill each other, stop and continue each other.
13. Write a program which acts as a chat application between two users on the same computer, using shared memory.
14. Setup SELinux on your virtual machine.

This list is a guideline. The instructor is expected to improve it continuously.

Departmental Electives

(CT-) Advanced Data Structures

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Apply software development life cycle in software industry.
2. Identify the importance of software requirements problem to understand the requirement management process.
3. Design and analyze effective use of UML using different design strategies.
4. Devise the procedure to assure the quality and maintainability of the product before and after deployment.
5. Summarize different testing strategies.

Unit I : Review of Basic Concepts: Abstract data types, Data structures, Algorithms, Big Oh, Small Oh, Omega and Theta notations, Solving recurrence equations, Master theorems, Generating function techniques, Constructive induction **[6 Hrs]**

Unit II : Advanced Search Structures for Dictionary ADT: Splay trees, Amortized analysis, 2-3 trees, 2-3-4 trees, Red-black trees, Randomized structures, Skip lists, Treaps, Universal hash functions, Trie ; **Hashing:** Simple tabulation hashing; chaining, dynamic perfect hashing, linear probing, cuckoo hashing **[10 Hrs]**

Unit III : Union Find Related Structures: Union-Find: Merging Classes of a Partition, Union-Find with Copies and Dynamic Segment Tree, List Splitting, Problems on Root-Directed Trees, Maintaining a Linear Order **[6 Hrs]**

Unit IV : Data Structures for Partition ADT: Weighted union and path compression, Applications to finite state automata minimization, Code optimization **[6 Hrs]**

Unit V : Data Structure Transformations: Making Structures Dynamic, Making Structures Persistent **[4 Hrs]**

Unit VI : Computational Geometry: Geometric data structures, Plane sweep paradigm, Convex Hull Different Paradigms and Quickhull , Dual Transformation and Applications , Lower Bounds on Algebraic tree model , Point Location and Triangulation **[8 Hrs]**

Text Books:

- Introduction to Algorithms; 3rd Edition; by by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein; Published by PHI Learning Pvt. Ltd. ; ISBN-13: 978-0262033848 ISBN-10: 0262033844
- Algorithms; 4th Edition; by Robert Sedgewick and Kevin Wayne; Pearson Education, ISBN-13: 978-0321573513
- Advanced Data Structures, Peter Brass, Cambridge University Press, ISBN-13: 978-0521880374

Reference Books:

- Algorithms; by S. Dasgupta, C.H. Papadimitriou, and U. V. Vazirani; Published by Mcgraw-Hill, 2006; ISBN-13: 978-0073523408 ISBN-10: 0073523402
- Algorithm Design; by J. Kleinberg and E. Tardos; Published by Addison-Wesley, 2006; ISBN-13: 978-0321295354 ISBN-10: 0321295358
- Pankaj Jalote, “An Integrated Approach to Software Engineering”, Narosa publication house.
- Fred Brooks, “Mythical Manmonths”, www.cs.drexel.edu/~yfcai/CS451/.

(CT) Advanced Data Structures Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Continuous evaluation: 50 Marks
End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Implement advanced data structures as abstract data types.
2. Write software for real life problems using advanced data structures.

List of Assignments:

1. Implement abstract data type for a dictionary as specified by the instructor.
2. Implement ADT for a partition.
3. Implement ADT for priority queue.
4. Implement an online path finding algorithm, which given dynamically changing edge weights suggests all possible paths, including the shortest path between any pair of vertices.
5. Write a program to find the convex hull of a set of points.

This is a suggested list. The instructor is expected to improve it continuously.

(CT) Advanced Microprocessors

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Analyze the protected mode, privilege levels, various descriptors and support for debugging
2. Explain Inter privilege level access mechanism, Multitasking support and paging facility.
3. Describe interrupt handling mechanism, use of MTRRs and PAT
4. Identify features of MMX technology, SIMD Execution Model and virtualization support

Unit I : System Architecture Overview: Support for operating-system and system-development software. Multitasking capability, subtasks and modularity in entire system, support offers multiple modes of operation, protection amongst OS and application programs, IA-32 architecture, Intel 64 architecture, System-Level Registers and data structures in protected mode, Global and Local Descriptor Tables **[6 Hrs]**

Unit II : Single-level Task: Protection mechanism and privilege in protected mode, IA-32 architecture, Debugging registers, memory management through segmentation and paging, support registers. **[6 Hrs]**

Unit III : Multilevel Tasks: Inter privilege level access mechanism call gate. Multitasking support, task switching and task gate. **[7 Hrs]**

Unit IV : Interrupts and Memory cache control: Interrupt, exception, faults, traps, interrupt handling in protected mode IDT, interrupt gate, trap gate, interrupt handling in protected mode, V86 mode. Extended features of V86, System Management Mode, Memory cache control: caching terminology, memory types and memory type range registers (MTRRs); Page Attribute Table (PAT), assigning memory types to regions of physical memory based on linear address mappings **[8 Hrs]**

Unit V : MMX and Streaming SIMD extensions: MMX technology, SIMD Execution Model, handling out-of-range conditions, execution environment for the SSE, SSE2, Streaming SIMD Extensions 3(SSE3), Supplemental Streaming SIMD Extensions 3 (SSSE3) and SSE4 **[7 Hrs]**

Unit VI : VMX Support: Intel Hyper-Threading Technology, Multi-core technology, Intel 64 architecture features, Intel Virtualization Technology **[6 Hrs]**

Text Books:

- Liu & Gibson, Microcomputer Systems, PHI, ISBN: 978-81-203-0409-3
- Barry B. Brey, The INTEL Microprocesoors, PHI, ISBN-81-203-1220-1

References:

- Tom Shanley, Protected Mode Software Architecture MINDSHARE, INC. Addison-Wesley Publishing Company, ISBN: 0-201-55447-X (.pdf)
- Intel® 64 and IA-32 Architectures Software Developer's Volumes 1, 2A, 2B, 3A, 3B (.pdf)

(CT) Advanced Microprocessors Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Term Work: 50 Marks
Oral: 50 marks

Course Outcomes:

Students will be able to

1. Demonstrate programming in protected mode.
2. Develop a program using SIMD instructions

List of Assignments:

1. Write an Assembly program to write (store) a string in Video RAM with help of BIOS Interrupts & display the written string on terminal along with the address of written string.
2. Write an Assembly program to write (store) a string in Video RAM without using BIOS Interrupts & display the written string on terminal along with the address of written string.
3. Write an Assembly program to accept any key from user & display the value of key pressed.e.g. input
   –a desired output
   ---"The key entered is 'a' "
4. Write a boot loader program. Execute the program on QEMU emulator.
5. Write an Assembly program for boot loader. Display a string "My OS".
6. Write a boot loader which will move from real mode to protected mode.
   a) Display a string in real mode(e.g "in real mode ")
   b) on pressing a key from keyboard transit from real mode to protected mode (display msg "in protected mode")
7. Prepare a CASE STUDY on Emulator.
8. Programming Assignments on MMX/ SSE/ SSE2 etc.

This list is a guideline. The instructor is expected to improve it continuously.

(CT) Web Systems and Technologies

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Analyze basic protocols used in World Wide Web.
2. Write and analyze behavior of web pages using HTML and CSS.
3. Write client side programming using appropriate technology.
4. Write server side programming using PHP.
5. Use different technologies for storing and transferring small database over the web.
6. Create, publish and test web services

Unit I : Web Essentials: Clients, servers, communication, basic Internet protocols, HTTP Request message, HTTP response message, web clients, generations of web applications. **[6 hrs]**

Unit II : Markup languages: HTML: fundamental HTML elements, head, body etc., basic XHTML syntax and semantics, document publishing; CSS: introduction, features, syntax, style properties of text, box, layout, list, table, cursor etc., user defined classes, inheritance. **[6 hrs]**

Unit III : Client-Side Programming: JavaScript: basic syntax, variables and data types, statements, operators, literals, functions; Javascript Objects: properties, references, methods, constructors, arrays, other built-in objects, debugging, host objects, document object model (DOM), document tree, DOM event handling, browsers Mobile Applications and Clients, Progressive Web Applications **[10 hrs**]

Unit IV : Server-Side Programming: PHP: client request, form data, request headers, server response, HTTP status codes, HTTP response headers, sessions, cookies, URL rewriting, separating programming and presentation, connection to databases. **[8 hrs]**

Unit V : Representing Web Data: XML: namespaces, DOM based XML processing, XSL, X Path, XSLT; AJAX: overview, basics, toolkits, security. **[6 hrs]**

Unit VI : Web Services: basic concepts, creating, publishing, testing and describing a web service, WSDL, XML services, communicating object data: SOAP, REST. **[4 hrs]**

Text Books:

- Jeffrey C.Jackson, "Web Technologies : A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035.

Reference Books:

- Marty Hall, Larry Brown,"Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978-0130897930.
- Robert. W. Sebesta, "Programming the World Wide Web", Forth Edition, Pearson Education, 2007, ISBN 978-0321489692.
- H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.

Online References

- https://www.w3.org/html/
- HTML, The Complete Reference http://www.htmlref.com/
- http://w3schools.org/
- http://php.net/
- https://jquery.com/
- https://developer.mozilla.org/en-US/docs/AJAX
- http://www.tutorialspoint.com/css/

(CT) Web Systems and Technologies Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Continuous evaluation: 50 Marks
Practical Exam:50 Marks

Course Outcomes:

Shared with the theory course: “Web Systems and Technologies”

Suggested List of Assignments:

1. Install, configure, compare and discuss features of 3 open source web servers.
2. Design fully functional website with attractive UI using the HTML, CSS and JS. It shall accept the form filled by user and check for syntax validity of every field.
3. Develop interactive multiple-choice quiz using HTML, JavaScript, AJAX and PHP.
4. Create a website to track session activities. Observe request and response objects.
5. Create a website where user can view some data but cannot view its true URL.
6. Case study: Configure any open source content management system (CMS) tool.

This list is a guideline. The instructor is expected to improve it continuously.

(CT -) Graphics and Multimedia

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Categories and compare various graphics drawing algorithms, 2D-3D transformations and polygon functions.
2. Classify basic graphics principles which are used in games, animations and film making
3. Estimate the components in multimedia system design (image, video, audio, etc)
4. Analyze and categories various software programs used in the creation and implementation of multi-media(interactive, audio, video, presentation, etc.)

Unit I : Introduction: Introduction to computer graphics, lines, line segments, vectors, pixels and frame buffers, vector generation, DDA and Bresenham's line and circle drawing algorithms, anti-aliasing, polygon representation, entering Polygons, Polygon filling: Seed fill, Edge fill, scan conversion algorithm. **[6 Hrs]**

Unit II : Transformations: Introduction, matrices, homogeneous coordinates, Basic 2D transformation like Scaling, Rotation, Translation, reflection etc 3-D Transformations: 3-D geometry, primitives, transformations, Rotation about an arbitrary axis, Concept of parallel and perspective projections, Viewing parameters, 3D viewing transformations. A brief introduction to hidden surface removal algorithms and Fractals. **[8 Hrs]**

Unit III: Segments and Animation: Introduction, segment table, segment creation, closing, deletion, renaming. Image transformations, raster techniques, Devices for producing animation, computer assisted animation, video formats, real time animation, frame-by-frame animation, method for controlling animation, animation software. **[6 Hrs]**

Unit IV :Multimedia System Design: Multimedia basics, Multimedia applications, Multimedia system architecture , Evolving technologies for multimedia , Defining objects for multimedia systems , Multimedia data interface standards , Multimedia databases. **[6 Hrs]**

Unit V :Multimedia File Handling: Compression and decompression , Data and file format standards , Multimedia I/O technologies, Digital voice and audio, Video image and animation , Full motion video, Storage and retrieval technologies. **[6 Hrs]**

Unit VI: Hypermedia: Multimedia authoring and user interface, Hypermedia messaging, Mobile messaging, Hypermedia message component , Creating hypermedia message, Integrated multimedia message standards, Integrated document management, Distributed multimedia systems. **[8 Hrs]**

Text Books:

- D. Hearn, M. Baker, “Computer Graphics – C Version”, 2nd Edition, Pearson Education,
- J. Foley, Van Dam, S. Feiner, J. Hughes, “Computer Graphics Principles and Practice”, 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9
- Ze-Nian Li, Mark S. drew, “ Fundamentals of Multimedia “, Pearson education, ISBN 81-7758-823-0

Reference Books:

- D. Rogers, “Procedural Elements for Computer Graphics”, 2nd Edition, TATA Mc-Graw-Hill Publication, 2001, ISBN 0 – 07 – 047371 - 4
- F. Hill, “Computer Graphics: Using OpenGL”, 2nd Edition, Pearson Education, 2003 ISBN 81– 297 – 0181 – 2
- S. Harrington, “Computer Graphics”, 2nd Edition, McGraw-Hill Publications, 1987 ISBN 0 – 07 – 100472 – 6
- G.S. BALUJA “Computer Graphics and multimedia “, DHANPAT Rai and Co.
- Rajan Parekh , “ Principles of multimedia : Tata McGraw-Hill, ISBN 978-0-07-058833-2

(IT-) Graphics and Multimedia Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Continuous evaluation: 50 Marks
End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to :

1. Apply various algorithms for generating and rendering graphical figures
2. Demonstrate computer graphics animation using latest animation software with improving self-learning ability.
3. Implement fundamental concepts in multimedia that are useful in design on modern information systems based on multimedia
4. Analyze and categories various software programs used in the creation and implementation of multi-media(interactive, audio, video, presentation, etc.).

List of Assignments:

1. Write a program to draw a line using DDA and Bresenham's line drawing algorithm.
2. Implement Mid-point circle drawing algorithm and understand the utilization of 8-way symmetry technique used for arc drawing
3. Draw the polygon using functions and implement basic 2D transformation.
4. Execute 3-D transformations (translation, scaling and rotation) for 3d images.
5. Write a program to generate animation effect.
6. Study of authoring tool – Director 8, to create presentation using multimedia files.
7. Parsing WAV sound files and reading it by programming in C/VC++.
8. Designing Media player using MCI commands to play sound – WAV, MIDI, AVI files etc.
9. Understanding standard Image file formats e.g. BMP, TIFF.
10. Implement Huffman coding algorithm for data compression.

(CT -) Fundamental of Digital Signal Processing

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Describe the components of DSP system, key DSP concepts and how do they relate to real applications
2. Represent discrete-time signals and systems analytically and visualize them in the time domain
3. Interpret the meaning and implications of the properties of signals, system and analyze the system in time domain
4. Represent and analyze signal and system in frequency domain
5. Utilize the z-transform to analyze discrete-time systems in terms of poles and zeroes
6. Implement digital filters in a variety of forms: direct form I and II, parallel, and cascade, and Window method design of different types of FIR filters(FIR)

Unit I : Introduction: Basic elements of digital signal processing (DSP) system, advantage of digital over analog signal processing, summary of DSP applications and introduction to DSP through these application. **[2 Hrs]**

Unit II : Signals And Systems :Basic concept of signals as array of values, standard signals, linearity, shift invariance, stability and causality, Linear Shift Invariant(LSI) systems ,I/O mapping and difference equations, Linear convolution, properties of linear convolution, computation of linear convolution ,A\D conversion process as sampling, Quantization, encoding, sampling theorem and anti aliasing filters. **[12 Hrs]**

Unit III : Analysis of Signals: Fourier transform, Fourier transforms of standard signals ,properties of Fourier transform, inverse Fourier transform, computation of Fourier transform, Discrete Fourier transform (DFT), DFT of standard signals , properties of DFT, computation of DFT, Fast Fourier Transform(FFT),Decimation In Time (DIT) and Decimation In Frequency(DIF), computation DIT/DIF FFTs, Inverse DFT & computation of IDFT using the FFT algorithms **[12 Hrs]**

Unit IV : Analysis of Signals: Analysis of LSI Systems: Magnitude / phase transfer function using Fourier transform, computation of transfer function, Z transform, Z transform of standard signals, properties of Z transform ,inverse Z transform computation of Z transform , System function from Z transform and pole-zero plots, computation of poles and zeros, Geometric constructs for transfer function viz. Region Of Convergence (ROC) using pole-zero plot and stability analysis. **[6 Hrs]**

Unit V : Digital Filters: Implementation of general difference equation, cascade and parallel forms of computation, Finite Impulse Response(FIR) and Infinite Impulse Response(IIR),filters from difference equations , FIR filter design using inverse Fourier transform and Windowing Gibb's phenomenon, computation of window, IIR filter design using impulse invariance and bilinear transform, computation of system function for given design parameters **[6 Hrs]**

Unit VI : DSP Processors : DSP micro-processor and their desirable features,ADSP-21XX and ADSP-210XX series of DSP micro-processor and their architectural features, implementing filters and FFTs on DSP microprocessor. Application of DSP: A brief overview of application of DSP in speech and image Processing **[2 Hrs]**

Text Books:

- J.G. Proakis , D.G. Manolakis , " Digital Signal processing" , 4th edition, Pearson Education, ISBN 0131873741
- Sanjit k. Mitra "Digital signal Processing a Computer based Approach" 3rd edition, Tata McGraw- Hill, ISBN 0070124467-0

Reference Books:

- Emmanuel C. Ifeachor , "Digital Signal Processing : A practical Approach", 4th edition,Addison Wesley , ISBN 0-0201-54413
- Salivahanan " Digital signal Processing" 2nd edition,Tata Mcgraw Hill , ISBN 978-0-07-066924-6
- Ashok Ambardar "Digital signal processing: A modern introduction", 1st edition,Cenage learning, ISBN 978-81-315-0837-4

( ) Fundamental of Digital Signal Processing Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/week

Examination Scheme:
Term Work – 50 marks
Oral – 50 marks

Course Outcomes:

Students will be able to:

1. Prepare comparative study report of scientific computing tools such as scilab/matlab
2. Analyze the response of a system using convolution, difference equation
3. Analyze the frequency response of the system
4. Design FIR filter

List of Assignments

1) Generate following discrete signals
   - Continuous sinusoid
   - Discrete sinusoid
   - Impulse signal
   - Unit step
   - Real exponential
   - Complex exponential
   - Triangular
   - Ramp.
2) Convolution

   a) Write a program to perform convolution of two finite length causal sequence b) Verify your answer using Conv function MATLAB

   c) Perform the convolution using filter function
3) Given a second /third order difference equation, write a program to

   Find out iteratively first n values of output. Accept n from the user.
4) Given a first order difference equation. E.g. y[n]-0.5y [n-1]=x[n]. The system is initially at rest.
   a) Write your own program to find out unit impulse and unit step response of the system and plot it

b) b) Find out unit impulse and unit step response of the system using filter function and plot it.

c) find out unit impulse response of the system using filter function and unit Step response using convolution and plot it .

Comment on the result.

5) Noise filtering by n point moving average

Generate a discrete sinusoid e.g s[n] = 2[n(0.9)n
Generate a noise signal using rand function. Superimpose this noise signal on discrete signal by doing addition. This is corrupted signal. Plot these three signals on the same graph using plot function by passing different line spec arguments. Now accept n from the user. Using n point averaging on corrupted signal, remove the noise. Plot the filtered signal. Change values of n. see the result. Comment on it.

6) Given the two finite length causal sequence x and y. Find out the autocorrelation of signal x and cross correlation of signal x and y. use the fliplr and conv function for this. Delay the signal e.g. by 4 units. I.e. X[n-4]. Perform the autocorrelation of signal x[n] with its delayed version. Plot it. Comment on the result.

7) Write a program to do circular convolution of two periodic signals x[n].

8) Consider a sinusoidal signal e.g. X(t)= Cos200∏t. Take a sampling frequency which satisfies Nyquist criteria.
   a) Generate Discrete time signal for n=0 to say 25.
   b) Accept N (no. of DFT points from the user)
      Take N greater than signal length. Pad remaining zeros to signal and find out the spectra using function FFT ( ). Plot the magnitude and phase plot for this.
   c) Reconstruct the signal x(n) from spectra using ifft()
   d) As N is greater than signal length, increase the signal length at given Sampling frequency by increasing n. (so that n will be equal to N) Again find the spectra and reconstruct the signal using ifft().
      Comment upon the reconstruction of signals in both cases

9) To do linear convolution and periodic convolution using FFT.

Take two finite equal length sequences. Find the linear and periodic convolution using FFT .

10) Write a program to accept pole zeros from the user. Plot those using function Zplane. Determine its rational Z form using function zp2tf.

11) Given the Z transform in rational form, determine the partial fraction expansion using residuez function. Using the same function covert the Z transform given in partial fraction expansion to rational form.

12) Finding frequency response of a system
    a) Find the frequency response of a linear system which is described by constant coefficient difference equation using freqz function
    b) Find the impulse response of a single pole filter using impz function.
13) Design of FIR lowpass, Highpas, bandpass and bandstop filter using rectangular window for given passband, stop band ripple, transition width and cutoff frequency.

This is a suggested list of assignments. The instructor is free to frame his/her own list of assignments.

(CT -) Advanced Data Structures (Honours course)

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

5. Understand operations associated with advanced data structures such as priority queues, dictionary structures, multi-dimensional data structures etc.
6. Analyze the time and space complexity of the operations associated with the advanced data structures and there by appreciate the use of these structures
7. Use advanced data structures to solve real life problems

Unit I : Review of Basic Concepts: Abstract data types, Data structures, Algorithms, Big Oh, Omega and Theta notations, Solving recurrence equations, Amortized complexity **[4 Hrs]**

Unit II : Priority Queues: Leftist Heap, Skew Heaps, Binomial, Fibonacci and Pairing Heaps, Double ended priority queues **[6 Hrs]**

Unit III: Dictionary Structures: Hash Tables, Universal hash functions, Balanced Binary Search Trees, Splay Trees, 2-3 trees, 2-3-4 trees, Red-black trees, Skip lists, Randomized Dictionary Structures, Treaps. **[6 Hrs]**

Unit IV : Multidimensional and Spatial Structures: Interval, Segment, Range, and Priority Search Trees, Quadtrees and Octrees, R-trees. **[8 Hrs]**

Unit V : Miscellaneous Topics: Persistent Data Structures, Cache-Oblivious Data Structures **[8 Hrs]**

Unit VI: Applications: IP Router Tables, Data Structures in Web Information Retrieval, Computational Biology, Geographic Information Systems, Computational Geometry: Geometric data structures. **[8 Hrs]**

Text Books:

- Advanced Data Structures; by Prof Peter Brass; Cambridge University Press; ISBN-10: 1107439825; ISBN-13: 978-1107439825

Introduction to Algorithms; 3rd Edition; by by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein; PHI Learning Pvt. Ltd.; ISBN-10: 0262033844; ISBN-13: 978-0262033848

Reference Books:

Handbook of Data Structures and Applications; by Dinesh P. Mehta (Editor), Sartaj Sahni (Editor); Chapman and Hall/CRC; ISBN-10: 1584884355;ISBN-13: 978-1584884354

Internet Resources:

MIT OpenCourseWare
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-851-advanced-data-structures-spring-2012/index.htm

COP 5536: Advanced Data Structures: Prof. Sartaj Sahni, University of Florida
https://www.cise.ufl.edu/~sahni/cop5536/