SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Syllabus for

M.Tech.(Computer Science)

(Applicable from the batch 2022–23 onwards)

Prasanthi Nilayam – 515 134

Sri Sathya Sai District, Andhra Pradesh, Ph: (08555) 287239, Fax: 286919

Website: www.sssihl.edu.in ; Email: registrar@sssihl.edu.in

REGISTRAR
Sri Sathya Sai Institute of Higher Learning
(Deemed to be University)
Vidyagiri, Prasanthi Nilayam
Sri Sathya Sai District, A.P. - 515 134
India

17-10-2022

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Syllabus for M.Tech. (Computer Science)

(Applicable from the batch 2022–23 onwards)

Programme Specific Objectives

1. To prepare graduates who will be successful professionals in IT industry, government, academia, research, entrepreneurial pursuit and consulting firms
2. To develop problem solving skills through regular interactions and training with Industry experts who bring current trends and industry requirements to discussion forums
3. To prepare graduates who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, research, design and implementation skills
4. To prepare graduates who will contribute to society as broadly educated, expressive, ethical and responsible citizens
5. To prepare graduates who will pursue higher goals of life-not succumb to mundane pressures of life by having clarity in thought, word and deed

Programme specific Outcomes

1. an ability to apply knowledge of mathematics, computer science and programming in practice
2. an ability to identify, critically analyze, formulate and solve engineering problems with comprehensive knowledge in the area of specialization
3. an ability to contribute by research and innovation to solve engineering problems
4. an ability to come up with a model and evaluation criteria so that the results/solution can be interpreted to provide for well-informed conclusions
5. an ability to understand the impact of engineering solutions in a contemporary, global, economic, environmental, and societal context for sustainable development
6. an ability to function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with positive attitude

Basic Structure

M.Tech.(Computer Science) is a four-semester programme for 72 Credits. Students with either Bachelor's degree in Engineering or Master's degree in Science are admitted. The students are expected to have undergone formal educational training with appropriate credits in first level courses in Computer organization and architecture, Computer networks, Database systems, Systems programming. Students are also expected to have undergone formal training in Programming with C, C++ and Java language. These courses constitute the prerequisites for the M.Tech.(Computer Science) programme. In addition to this, fair knowledge of Operating systems, Compiler Design, Formal Languages and Automata Theory, hands on experience with UNIX environment and familiarity in programming with PYTHON, MATLAB etc. will be advantageous to the students seeking admission to this course.

A student takes in all 10 courses of which 6 are core courses which pertain to the fundamentals of computer science. The remaining 4 are elective courses which give the scope for specialization in the individual's interest or thrust areas of the department. Most of the courses have a complementary hands-on lab course associated with it.

A Student undertakes a Mini-project worth 2 credits in second semester and a Project worth 18 credits in third and fourth Semesters. This gives the student a firm training in designing and programming for a solution in the areas of interest.

Special components such as awareness courses and Problem-Solving Lab(2 credits) give much needed focus and training in developing soft-skills, life-skills etc.

Course Design

Semester Duration: A maximum of 14 weeks is typically used in the semester to deliver any subject.

Credit Distribution:

1. Total 72 credits for the M.Tech(CS) program is distributed as given below:

a.	6 Core Theory subjects	18 (3 credits each)
b.	5 Core Practical subjects	10 (2 credits each)
c.	4 Elective subjects	16 (4 credits each)
d.	2 Seminar presentations	2 (1 credit each)
e.	1 mini-project during second semester	2 credits
f.	Problem-Solving Lab	2 credits
g.	Project during the third and fourth semester	18 credits
h.	Awareness Courses	4 credits
	Total	72 Credits

2. An elective subject may be designed to have two different components/mode of delivery namely, Theory and Practical with clear division of credits among the components. For example, a 4 credit course could be specified as 2L+2P to indicate 2 credits for lectures and 2 credits for practical or as 3L+P meaning 3 credits for lecture and one credit for practical or L+3P to indicate 1 credit for lectures and 3 credits for practical.

3. Typically, one lecture credit (T) is given one period per week, one practical credit (P) is given minimum of two and maximum of three periods per week.

4. In the first semester students have two seminar presentations that demand independent study of research papers, latest technology trends in the areas of interest and presentation skills. Seminar-I will be on the study of Research Papers published in journals. Seminar-II will be based on happenings and latest technology trends in the area of Computer Science through practical demonstrations by the candidate.

5. A mini-project worth 2 credits during the second semester should be undertaken.

6. The students have to do a two-semester substantial project worth 18 credits starting from the third semester.

7. A problem solving Lab(2 credits) is introduced in the 4th semester in which the students will undertake exercises to focus on problem solving skills. This will be driven by both faculty and experts from industry who will expose the students to certain real-world scenarios where their skills learnt during the programme can be applied.

* * *

DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

SCHEME OF INSTRUCTION AND EVALUATION

M.Tech.(Computer Science)

(Effective from the batch 2022-23 onwards)

Paper Code	Title of the Paper	Credits	Hours	Types of papers	Modes of Evaluation	Maximum Marks
FIRST SEMESTER						
MTCS-101	Design and Analysis of Algorithms	3	3	T	IE2	100
MTCS-101(P)	Practicals: Design and Analysis of Algorithms	2	4	P	I	50
MTCS-102	Advanced Computer Architecture	3	3	T	IE2	100
MTCS-102(P)	Practicals: Advanced Computer Architecture	2	4	P	I	50
MTCS-103	Parallel Processing	3	3	T	IE2	100
MTCS-103(P)	Practicals: Parallel Processing	2	4	P	I	50
MTCS-104	Elective-I	4*	4*	T	IE2	100*
MTCS-105	Seminar-I	1	1	-	I	50
MTCS-106	Seminar-II	1	1	-	I	50
MAWR-100	Awareness Course-I: Fundamentals of Indian Culture	1	1	T	I	50
		--- 22	28		------	700
SECOND SEMESTER						
MTCS-201	Theory of Computation	3	3	T	IE2	100
MTCS-202	Distributed Systems	3	3	T	IE2	100
MTCS-202(P)	Practicals: Distributed Systems	2	4	P	I	50
MTCS-203	Topics in Database Management Systems	3	3	T	IE2	100
MTCS-203(P)	Practicals: Topics in Database Management Systems	2	4	P	I	50
MTCS-204	Elective-II	4*	4*	T	IE2	100*
MTCS-205	Elective-III	4*	4*	T	IE2	100*
MTCS-208	Mini-Project	2	4	MP	I	50**
MAWR-200	Awareness Course-II: Sources of Values	1	1	T	I	50
		--- 24*	30*		------	700
THIRD SEMESTER						
MTCS-301	Elective-IV	4*	4*	T	IE2	100*
MTCS-401	Project Work – Review	-	22	PW	-	50***
MAWR-300	Awareness Course-III: Work Culture, Ethics and Values	1	1	T	I	50
		--- 5	27*		-------	200
FOURTH SEMESTER						
MTCS-401	Project Work	18	24	PW	E2	150***
MTCS-402	Problem-Solving Lab	2	2	P	I	50****

MAWR-400	Awareness Course-IV:	1	1	T	I	50
	SSSIHL's Core Values and Philosophy	-- 21	27			-----250
	TOTAL:	72	112*			1850*

PS: Please refer to guidelines for 'Modes of Evaluation for various types of papers', and 'Viva voce nomenclature & scope and constitution of the Viva voce Boards'.

* Credits split between Lectures and Practical, total marks for the subject, and the grand total marks for the paper, may change based on the credits allocated for the Lecture and Practicals of the elective(s) the students opt for. i.e., the elective paper A may be designed to have two different components/mode of delivery namely, Theory and Practical with clear division of credits among the components. For example a 4 credit course could be specified as 2L+2P to indicate 2 credits for lectures and 2 credits for practical or as 3L+1P meaning 3 credits for lecture and one credit for practical or 1L+3P to indicate 1 credit for lectures and 3 credits for practical.

** The Mini-Project in the second semester will be undertaken during the second semester by the candidate. This could be based on an internship (taken online/on-campus) with an industry or a field work etc. with a mentoring faculty from the department. It is recommended that for CS related Mini-Project, performance sensitive & versatile environments/platforms such as C, C++, Java, C# etc be used so that students will develop strong programming skills. Students will be asked to make a presentation along with a submission of the report of work done so far which will be evaluated internally by a panel of minimum two faculty of the department constituted by HoD/Associate HoD. Total marks for the mini-project would be for 50 marks which will be equally distributed between the presentation and the report submitted.

With regard to the final year project, the Project Work topic would be finalized by the end of the second semester, and the Project Work starts in the third semester and gets completed in the fourth semester. The interim review would consist of an oral examination to assess the progress made by the student in the project work. Students will be asked to make a presentation along with a submission of a report of work done so far.

*** Total marks for the Project Work would be for **200 marks**, which would include

- **50 marks** for the review of the project work by the student at the end of the third semester (please see **)
- **100 marks** for the Project Report Examination
- **50 marks** for Project Viva voce conducted at the end of the 4th semester.

**** Total marks for the Problem-Solving lab would be for 50 marks. This is done internally with serial evaluations(SEs) and a Summative test(ST). It will be handled by industry experts and faculty to train the students with necessary design thinking and hands-on skill for solving programming type problems so that they become industry ready.

Continuous Internal Evaluation (CIE) & End Semester Examination (ESE)

Indicator	Legend
IE1	CIE and ESE ; ESE single evaluation
IE2	CIE and ESE ; ESE double evaluation
I	Continuous Internal Evaluation (CIE) only Note: 'I' does not connote 'Internal Examiner'
E	End Semester Examination (ESE) only Note: 'E' does not connote 'External Examiner'
E1	ESE single evaluation
E2	ESE double evaluation

Indicator Legend

Indicator	Legend
T	Theory
P	Practical
V	Viva voce
PW	Project Work
D	Dissertation
MP	Mini-Project

Note: The choice of electives being offered in each semester is at the discretion of the Head of the Department.

STREAMS of Elective Courses

STREAM I: INTELLIGENT SYSTEMS AND KNOWLEDGE ENGINEERING

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
ISKE 1	Artificial Intelligence	T	3	Good Analytical Skills, Basic knowledge of Probability and Statistics modeling
ISKE 1(P)	Practicals: Artificial Intelligence	P	1	Good programming skill, Discrete Maths
ISKE 2	Natural Language Processing	T	3	First level Course in A.I
ISKE 2(P)	Practicals: Natural Language Processing	P	1	First level Course in A.I

ISKE 3	Machine Learning	T	2	Foundations in Probability and Statistics.
ISKE 3(P)	Practicals: Machine Learning	P	2	Foundations in Probability and Statistics.
ISKE 4	Mining of Big Data Sets	T	2	Basics in Algorithms, probability theory, and linear algebra.
ISKE 4(P)	Practicals: Mining of Big Data Sets	P	2	Data structures, Good programming skills
ISKE 5	Deep Learning	T	2	Matrix arithmetic, probability, Calculus
ISKE 5(P)	Practicals: Deep Learning	P	2	Data structures, Good programming skills
ISKE 6	Fundamentals of Blockchain Technologies and Applications	T	3	Algorithms, Data structures
ISKE 6(P)	Practical: Fundamentals of Blockchain Technologies and Applications	P	1	Data structures, Good programming skills

STREAM II: ADVANCED COMPUTER NETWORKS

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
ACN 1	Wireless and Mobile Networks	T	3	First level Course in Computer Networks
ACN 1(P)	Practicals: Wireless and Mobile Networks	P	1	First level Course in Computer Networks

STREAM III: HUMAN COMPUTER INTERACTION

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
HCI 1	Digital Image Processing	T	3	Basic probability, statistics and Linear Algebra
HCI 1(P)	Practicals: Digital Image Processing	P	1	Data structures, Good programming skills
HCI 2	Medical Image Processing	T	3	First level Course in Image Processing
HCI 2(P)	Practicals: Medical Image Processing	P	1	First level Course in Image Processing
HCI 3	Computer vision	T	3	First level Course in Image Processing
HCI 3(P)	Practicals: Computer vision	P	1	First level Course in Image Processing

STREAM IV: THEORETICAL COMPUTER SCIENCE

Paper Code	*Elective Title*	Theory/ Practical	Credits	*Prerequisite*
TCS 1	Advanced Algorithms	T	3	First level Course in Algorithms, Probability.
TCS 1(P)	Practicals: Advanced Algorithms	P	1	First level Course in Algorithms, Probability.
TCS 2	Cryptography	T	3	Basic Number theory
TCS 2(P)	Practicals: Cryptography	P	1	good skills in programming
TCS 3	Design of Quantum Algorithms	T	3	First level Course in Algorithms, Probability.
TCS 3(P)	Practicals: Design of Quantum Algorithms	P	1	First level Course in Algorithms, Probability.

STREAM V: COMPUTER SYSTEMS

Paper Code	Elective Title	Theory/ Practica l	Credits	Prerequisite
CS 1	Advanced Programming in the Unix Environment	T	2	First Level Course in OS
CS 1(P)	Practicals: Advanced Programming in Unix Environment	P	2	Programming in C, C++

STREAM VI: MULTI-CORE AND PARALLEL COMPUTING

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
MPC 1	High Performance Computing with Accelerators	T	2	First Level course in Architecture, Systems Programming
MPC 1(P)	Practicals: High Performance Computing with Accelerators	P	2	First Level course in Architecture, Systems Programming
MPC 2	Cloud Computing	T	3	First Level course in Architecture, Systems Programming
MPC 2(P)	Practicals: Cloud Computing	P	1	First Level course in Architecture, Systems Programming

STREAM VII: SOFTWARE ENGINEERING

Paper Code	*Elective Title*	Theory/ Practical	Credits	*Prerequisite*

SE 1	Object Oriented System Design	T	3	First level course in Algorithms, Programming knowledge
SE 1(P)	Practicals: Object Oriented System Design	P	1	Good command over Programming, data structures

* * *

M.TECH. (COMPUTER SCIENCE)
CORE COURSES (4 CREDITS)

MTCS-101 DESIGN AND ANALYSIS OF ALGORITHMS

(3 Credits) (42 Periods)

Course Objectives:
The course will familiarize the students with tools for designing and analyzing algorithms. Assuming that the student has studied a basic course in algorithms, the course builds over deeper topics such as proof of correctness of algorithms, design through paradigms and probabilistic analysis. The course provides a broad range of algorithms relevant for the current trends.

Course Outcomes: At the completion of the course the student will be able to

- Know how to solve a problem and subsequently design an algorithm through an inductive process.
- Understand loop invariants and use them to prove the correctness of algorithms.
- Understand different paradigms for designing algorithms such as Divide and Conquer, Dynamic Programming and Greedy Method.
- Understand probabilistic and amortized analysis of algorithms
- Design randomized algorithms
- Know the analysis of various sorting algorithms
- Design solutions for graph problems and prove their correctness.
- Familiarize with specialized topics such as string matching and computational geometry.

Course Syllabus:

Unit 1: Design & Analysis Techniques (10 periods)
Growth of functions, Loop invariant, Divide and Conquer, Master method for solving recurrences, Dynamic programming, Greedy Algorithms (excluding starred sections), Amortized analysis, Probabilistic Analysis and Randomized Algorithms (excluding starred sections).

Unit 2: Sorting & Order statistics (6 periods)
Sorting in linear time, Heapsort, Quicksort, Medians and Order statistics

Unit 3: Data Structures
Elementary data structures, Hash tables, Binary search trees, Red Black trees, Augmenting Data Structure, Data structure for disjoint sets (12 periods)

Unit 4: Graph algorithms (10 periods)
Graph searching techniques, Minimum spanning trees, single source shortest paths, all pairs shortest paths

Unit 5: Specialized topics (4 periods)
String Matching, Computational geometry.

Total Periods: 42 periods

Reference Text: Thomas H Cormen, Charles E Leiserson, Ronald Rivest, Cliffor Stein., *Introduction to algorithms*, 3rd edition, Chapters: 1 to 17, 21 to 25, 32, 33, (July 31, 2009)

Suggested Reading: Anany Levitin., *Introduction to Design & Analysis of Algorithms*, 2nd edition, Year: 2012.

MTCS-101(P) Practicals: DESIGN AND ANALYSIS OF ALGORITHMS

(2 Credits)

Course Objectives:

Algorithms / Exercises from different units in the syllabus will be implemented in Lab. The student writes their programs in Python language.

Course Outcomes: At the completion of the lab course the student will be

- equipped with the skill set to prove the correctness through strategies such as loop invariants and bound functions.
- Able to write programs by the principles of algorithmic design.

Recommended Exercises:

- convert a recursive programme to an iterative programme
- write programs for various paradigms such as Divide and Conquer, Dynamic Programming and Greedy Method.
- analyze randomized algorithms
- code various sorting algorithms
- work on graph with both representations: adjacency matrix and list
- write code for various graph algorithms
- write code for geometric and string algorithms

Reference Text: Thomas H Cormen, Charles E Leiserson, Ronald Rivest, Cliffor Stein., *Introduction to algorithms*, 3rd edition, Chapters: 1 to 17, 21 to 25, 32, 33, (July 31, 2009)

Suggested Reading: Anany Levitin, *Introduction to design & analysis of algorithms*, 2nd edition, Year: 2012.

* * *

MTCS-102 Advanced Computer Architecture (3 Credits) (42 Periods)

Course Objectives: This course will introduce students to the advanced concepts of computer organization and architecture. It is to familiarize students about design principles of organization and micro architecture that gives high performance capability. The course also gives a broader perspective of modern processor designs such as super scalars, vector architecture, GPUs etc.

Course Outcomes: At the completion of the course the student will be able to

- know the classes of computers, and new trends and developments in computer architecture
- Understand pipelining, instruction set architectures, memory addressing.
- Understand the various techniques to enhance a processor's ability to exploit Instruction-level parallelism (ILP), and its challenges.
- Understand exploiting ILP using dynamic scheduling, multiple issues, and speculation.
- Understand the performance and efficiency in advanced multiple-issue processors.
- Understand symmetric shared-memory architectures and their performance.
- Understand multiprocessor cache coherence using the directory based and snooping class of protocols.
- Understand the several advanced optimizations to achieve cache performance.
- Understand virtual memory and virtual machines

Course Syllabus:

Unit 1: (7 periods)
Instruction Set Architectures, Microcode (H&P5 Chapter 1, H&P5 Appendix A)
Pipelining Review (H&P5 Appendix C) Cache Review (H&P5 Appendix B-1 - B-40)

Unit 2: (12 periods)
Instruction level parallelism – hardware and software techniques (e.g., dynamic scheduling, superscalar, static and dynamic branch prediction, VLIW, loop unrolling). (H&P5 Chapter 3)
Exceptions (H&P5 Appendix C)

Unit 3: (14 periods)
Branch Prediction (H&P5 Appendix C, Chapter 2) Memory hierarchy – advanced concepts in caches (e.g., prefetching, lockup-free caches, and multi-level caches),
main memory, and virtual memory. (H&P5 Chapter 2 (P.71 – P.105))
Memory Protection (H&P5 Appendix B(B41 -B67), Chapter 2 (P.105 – P.112))

Unit 4: (9 periods)
Vector Processors and GPUs (H&P5 Chapter 4) Multiprocessors/multicore – overview of different models, cache coherence with shared-memory systems/multicore (snoopy and directory solutions), synchronization. (H&P5 Chapter 5)

Total 42 Periods

Reference text:

1. [H&P5] "Computer Architecture: A Quantitative Approach (5th Edition)", John L. Hennessy and David A. Patterson, ISBN: 978-0123838728, 2012.

Suggested readings:

1. D. M. Harris and S. L. Harris (Morgan Kaufmann, 2012), *"Digital Design and Computer Architecture"*, 2nd edition.

2. John P. Shen and Mikko H. Lipasti, *"Modern Processor Design: Fundamentals of Superscalar Processors* (1st Edition)", 2004. ISBN: 0070570647, Princeton University Library Owns.

MTCS-102(P) Practicals: Advanced Computer Architecture (2 Credits) (56 periods)

Course Objectives: This lab course is expected to reinforce some of the important concepts and principles that are explained in MTCS-102. The main objective is to give hands-on training to check and verify some principles taught as part of theory.

Course Outcomes: At the completion of the course the student will be able to

- Install and run a simulator on a computer
- Understand the importance of simulation studies
- Understand various parameters such as instruction cycles, cycles per instruction, execution time etc for a specific benchmark/application program
- Understand the impact of cache configuration on variety of benchmarks
- Introduce new modules into the simulator to rebuild it with certain features pertaining to memory subsystem or CPU etc.

Recommended assignments:

1. Installing and getting acquainted with a computer system simulator such as Gem5
2. Running standard benchmark sets such as SPEC2000 and observing execution characteristics
3. Altering cache configurations to get the best performance for a certain benchmark/application program
4. Introducing a new module in the simulator for memory subsystem to introduce L3 cache
5. Introducing a new module in the simulator for designing multi-core CPU

Reference Material:

1. Gem5 Tutorials and Exercises Online.

* * *

MTCS-103 PARALLEL PROCESSING

(3 Credits) (42 Periods)

Course Objective: To introduce and develop understanding on the following topics:

- Parallelism, Parallel Computers and Parallel Programming
- Parallel Architecture, Interconnection Network and Classification. Distributed Memory and Shared Memory programming using MPI & OpenMP.
- Parallel Algorithms Design, Foster's Design Methodology
- Performance Analysis of Parallel Algorithms and Scalability

Course Outcome: Students will have comprehensive knowledge and develop basic skill on the following concepts:

- Types of Concurrency, recognizing Data Parallelism and Functional Parallelism
- Classification of Architecture
- Properties and classification of Interconnection Network Topologies
- Designing of parallel Algorithm using Foster's Design Methodology and analysis of it.
- Performance Analysis with Amdhal's Law, Gustaffson Barsis's Law, Karp Flat Metric
- Scalability of parallel systems and Iso-efficiency Analysis
- MPI Collective Calls, Barrier synchronization, blocking and Non-blocking Point to Point calls,
- Shared Memory Programming, Parallel Pragmas, number of threads in one program, race conditions, data & functional parallelism in OpenMP.

Course Syllabus:

Unit 0: Introduction - Modern Parallel Computers - Types of Concurrency – Programming. 3 Periods

Unit 1: Parallel Architecture – Interconnection Network – Processor arrays – Multiprocessors – Multi Computers – Flynn's taxonomy. 5 Periods.

Unit 2: Parallel Algorithm Design – Foster's Design Methodology – Example Problems. 4 Periods

Unit 3: Message Passing programming Model – MPI – Point to Point & Collective Calls. 4 Periods.

Unit 4: Algorithms for Illustrations – Sieve of Eratosthenes – Floyd's Algorithm. (To discuss all the concepts introduced so far). 4 Periods

Unit 5: Performance analysis – Speed up and Efficiency – Amdahl's Law – Gustafson's Barsis Law – Karp Flatt Metric – Isoefficiency Metric. 4 Periods

Unit 6: Matrix Vector Multiplication – Monte Carlo Methods – Matrix Multiplication – Solving linear System - finite Difference Methods - sorting algorithm - combinatorial Search. 14 Periods

Unit 7: Shared Memory Programming – Open MP. 4 Periods.

Total: 42 Periods.

Reference Text:

Sri Sathya Sai Institute of Higher Learning (Deemed to be University) PRASANTHI NILAYAM India

Michale J Quinn, *Parallel Programming in C with MPI and OpenMP*, Tata McGraw Hill 2004.

Suggested Readings:
Anantha Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing*, *Pearson education LPE*, 2nd Edition, 2004

MTCS-103(P) Practicals: PARALLEL PROCESSING (2 Credits)(56 periods)

Course Objectives: To give practical knowledge in:

Shared Memory programming using MPI & OpenMP.
Parallel Algorithms Design
Performance Analysis of Parallel Algorithms and Scalability

Course Outcome: At the completion of the course a student will be sufficiently skilled in parallelizing an algorithm/application using standard message passing libraries such as MPI, OpenMP.

1. Message Passing programming Model – MPI – Point to Point & Collective Calls.
2. Document classification Problem
3. Matrix Vector & Matrix Matrix Multiplication
4. Parallel Quicksort
5. Shared Memory Programming – Open MP

Reference Text:
Michale J Quinn, *Parallel Programming in C with MPI and OpenMP*, Tata McGraw Hill, 2004.

Suggested Readings:
Anantha Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing*, Pearson education LPE, Second edition, 2004.

* * *

MTCS-201 Theory of Computation

(3 Credits) (42 Periods)

Course Objectives:

- Learn several formal mathematical models of computation along with their relationships with formal languages
- Learn that not all problems are solvable by computers, and some problems do not admit efficient algorithms.
- Learn techniques to show that a problem is not efficiently solvable

Course Outcomes:

- Demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages.
- Understand that there are limitations on what computers can do, and learn examples of unsolvable problems.
- Appreciate that certain problems do not admit efficient algorithms, and identify such problems.

Course Syllabus:

Unit 1 Introduction to Basic Models of Computation and the finite representation of Infinite Objects — 4 periods

Unit 2 Finite Automata and Regular Languages — 6 periods

Unit 3 Pushdown Automata and Context - Free Language — 6 periods

Unit 4 Turing Machines and their Variants - Recursive Functions - Church's Thesis — 14 periods

Unit 5 Un-decidability - Reducibility and Completeness – Time Complexity and NP-Completeness — 12 periods

Total 42 periods

Reference Text:

Harry Lewis R, Christos H. Papadimitriou, *Elements of theory of computation*, 2nd edition, PHI Publications, 1998.

Coverage of Key Text Chapters: 1 (only 1.7 and 1.8), 2, 3, 4 (except 4.4 and 4.6), 5, 6, 7.

Suggested Readings:

John. C. Martin, *Introduction to Languages and the Theory of Computation*, Tata McGraw-Hill, 2003.

* * *

MTCS-202 DISTRIBUTED SYSTEMS

(3 Credits) (42 Periods)

Course Objectives:

This course deals with the theory and applications of building a large-scale distributed system. The course includes an overview of large scale systems in use today such as Youtube, Twitter, Amazon S3 and Netflix. Objective is to expose students to the salient features of building such large-scale systems in practice, starting from the bottom most layer of how to deploy computing resources for scale, all the way to running compute "jobs" that process Petabytes of data in a single job. The fundamentals of sharding as a technique for scaling, protocols for leader election such as Paxos, various methods of implementing Consistent Hashing, protocols and techniques for data replication, techniques for resource placement across availability zones, etc. will form the core part of this course.

Course Outcomes:

At the completion of the course the student will be able to

- Know the core ingredients and design principles involved in building large-scale systems
- Understand the design tools and how to reason about which design to apply and when
- Understand how large-scale distributed file systems are built
- Understand important techniques in building a reliable system
- Understand the principle of Consistent Hashing and how it can be applied in various scenarios
- Know the logical clocks and their role in Distributed Systems
- Know the principles of Cloud Computing
- Take an existing, stand-alone system and convert it into a large-scale distributed system

Course Syllabus:

Unit 1: Fundamentals---Remote method invocation, indirect communication, time and global states, replication (TEXT-I: Ch. 5, 6, 14, 18) 6 periods

Unit 2: Services: P2P systems, distributed file systems, distributed databases (including 2-phase commit, distributed locks) (TEXT-I: Ch. 10, 12, 17) 6 periods

Unit 3: Coordination and agreement in distributed systems (TEXT-I: Ch. 15) 2 periods

Unit 4: Consensus protocols---Paxos, Proof-work, Proof-of stake (Lamport 2001, Nguyen et al 2019) 4 periods

Unit 5: Isolation and consistency semantics (Coulouris: Chapters 16, 17; TEXT-II: Chapter 15) 2 periods

Unit 6: Large-Scale Distributed Systems: Youtube, Netflix, Twitter, Amazon (highscalability.com) 2 periods

Unit 7: Cloud Computing – Introduction, cloud data centers, Computing in the cloud and challenges, MapReduce paradigm (TEXT-II: Ch. 2-3, 5) 6 Periods

Unit 8: Sharding as a technique for scalability and reliability (Baguia and Nguyen, 2015) 2 Periods

Unit 9: Consistent Hashing and example applications (Karger et al, 1997) 2 Periods Unit 10: Building a high-performance key-value store (Kang et al, 2019) 2 Periods

Unit 11: Designing systems for disaster recovery (Lenk and Tai, 2014) 2 Periods

Unit 12: Distributed systems security: Encryption/decryption, Digital signatures, Needham-Schroeder protocol, Kerberos (TEXT-I: Ch. 11) 2 periods

Unit 13: Case studies: Google's distributed applications 4 Periods

Total: 42 Periods

REFERENCE TEXT:

1. Coulouris, Dollimore, Kindbeg and Blair, TEXT-I: *Distributed Systems – Concepts & Design*, 5th Edition, Addison-Wesley, 2012 [Chapters 5, 6, 10-12, 14-15, 16-18].
2. Kenneth P. Birman, TEXT-II: *Guide to Reliable, Distributed Systems*, Springer, 2012 [Chapters 2-3, 5, 15].

SUGGESTED READINGS:

Unit 4 references:

1. Leslie Lamport, Paxos made simple, 2001. https://lamport.azurewebsites.net/pubs/paxos-simple.pdf
2. Cong T. Nguyen, et al. Proof-of-stake consensus mechanism for future blockchain networks: Fundamentals, applications, and opportunities, IEEE Access, 2019.

Unit 6 references: highscalability.com 1. Youtube architecture http://highscalability.com/youtubearchitecture

2. Netflix architecture

http://highscalability.com/blog/2015/11/9/a-360-degree-view of-the-entire-netflix-stack.html

3. Twitter architecture http://highscalability.com/blog/2013/7/8/the-architecture twitter-uses-to-deal-with-150m-active-users.html

4. Amazon architecture http://highscalability.com/amazon architecture

Unit 8 references:

1. Sikha Bagui and Loi Tanh Nguyen, database sharding: To provide fault tolerance and scalability of big data on the cloud, IJCAC, 5(2), 36-52, 2015.

Unit 9 references:

1. David Karger et al., Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web, ACM STOC, 1997.

Unit 10 references

1. Yangwook Kang et al., Towards building a high performance, scale-in key-value storage system, ACM SYSTOR, 2019.

Unit 11 references:

1. Alexander lenk and Stefan Tai, *Cloud Standby: Disaster Recovery of Distributed Systems in the Cloud*, ESOCC 2014.

MTCS-202(P) Practicals: DISTRIBUTED SYSTEMS (2 Credits)

Objectives: This course introduces students to the practical aspects of the theoretical concepts, techniques and methods learnt.

Course Outcomes: At the completion of the course the student will be able to:

- Do programming that enables one to know the functioning of Internet Protocols, WWW, IPC
- Understand programming with MPI and solve problems
- Gain insight into programming with the help of RPC and RMI
- Learn group communications and processes and threads as visualized in Distributed environment

Suggested Experiments:

PART-I

* IPC and MPI can be programmed and demonstrated.

* RPC and RMI can be programmed and demonstrated.

* Communications between groups and approaches to shared memory

* Processes and Threads can be programmed and used in a distributed environment

* Learn Hadoop Framework and implement suitable exercises
* Programming atomic commit protocol Locks, Transactions and Time Stamp Ordering through simulations

PART-II : Hadoop Programming and SPARK

PART-III: Mini Project

Reference Text:

1. Coulouris, Dollimore, Kindbeg and Blair, *Distributed Systems – Concepts & Design*, 5th Edition, Addison-Wesley, 2012 [Chapters 5 –21]

* * *

MTCS-203 TOPICS IN DATABASE MANAGEMENT SYSTEM

(3 Credits) (42 Periods)

Prerequisites: A first level course in Database Management Systems.

Course Objectives:

- Introduce the internals of a database management system.
- Introduce concurrency and recovery of databases.
- Explain distributed and parallel databases.

Course Outcomes: At the end of the course, the student will be able to understand

- What happens inside a database when a query is submitted.
- How relational algebra operations are performed by the database engine.
- How queries can be optimized for quick execution.
- Maintaining ACIDity even in the presence of concurrent transactions.
- How a database can be recovered after a system crash.
- How the relational algebra operations can be parallelized.
- How concurrency of transactions can be achieved in a distributed system.

Syllabus:

Unit 1 : QUERY PROCESSING AND OPTIMIZATION 9 periods

Query Processing, Query Optimization.

Unit 2 : TRANSACTION MANAGEMENT 12 periods

Concurrency Control, Recovery System.

Unit 3 : PARALLEL AND DISTRIBUTED DATABASES 15 periods

Database-System Architectures, Parallel and Distributed Storage, Parallel and Distributed Query Processing, Parallel and Distributed Transaction Processing.

Unit 4 : ADVANCED TOPICS 6 periods

Advanced Indexing Techniques, Blockchain Databases.

Total 42 periods

Reference Text: Silberschatz, A., Korth, H. F., and Sudarsham, S. *Database System Concepts*, 7th Edition, McGraw-Hill (2020).

Chapters 15, 16, 18, 19, 20, 21, 22, 23, 24, and 26.

Suggested Readings:

1. Fundamentals of Database Systems by Elmasri, R., and Navathe, S. B., 4th edition, Pearson Education(2007).

2. Ramakrishnan, R., and Gherke, J. *Database Management Systems*, Second Edition, McGraw-Hill (2000).

3. Sunderraman, R.*Oracle 10g Programming: A Primer*, Addison-Wesley, (2008)

MTCS-203(P) Practicals: TOPICS IN DATABASE MANAGEMENT SYSTEMS

(2 Credits) (56 periods)

Course objectives: To provide the student with necessary hands on skill set for implementing some important concepts in the database management system. To make the student practically design and endeavor as a database system.

Course Outcomes: The student will be capable of:

- How to implement some relational algebra operations like join.
- How to program some internal processes like lock manager.

Recommended Assignments:

1. Retrieve evaluation plan of a query.
2. Implement a simple query processor.
3. Implement a simple Lock Manager
4. Implement External Sort.

Reference Text: Silberschatz, A., Korth, H. F., and Sudarsham, S. *Database System Concepts*, 7th Edition, McGraw-Hill,(2020).

MTCS-402 Problem-Solving Lab

(2 Credits) (28 Periods)

Course objectives: To provide the student with necessary design thinking and hands-on skill for solving programming type problems so that they become industry ready.

Course Outcomes: The student will be:

- Able to solve types of problems for preparing towards technical interviews
- Able to develop time and space efficient algorithms for industry level projects/applications

Syllabus: Exercises can be coined from industry based problems or from social platforms such as leetcode.com, hackerrank.com etc. Example list of medium and high standard problems listed below

Sliding window : K-diff-Pairs-in-an-Array,Valid-Triangle-Number, Binary-Subarrays-With-Sum,Frequency-of-the-Most-Frequent-Element, Longest-Substring-with-At-Least-K-Repeating-Characters

Binary Search : Median-of-Two-Sorted-Arrays , Find-Minimum-in-Rotated-Sorted-Array, Find-Minimum-in-Rotated-Sorted-Array, Search-in-Rotated-Sorted-Array

Hash Map : Group-Anagrams, Max-Points-on-a-Line , Is-Subsequence , H-Index

Trees : Find-Mode-in-Binary-Search-Tree, Quad-Tree-Intersection, Maximum-Width-of-Binary-TreeClosest-Leaf-in-a-Binary-Tree, Sum-of-Distances-in-Tree

Graphs: Tree-of-Coprimes, Longest-Subsequence-Repeated-k-Times, Number-of-Valid-Move-Combinations-On-Chessboard, Maximum-Path-Quality-of-a-Graph.

Sri Sathya Sai Institute of Higher Learning (Deemed to be University) PRASANTHI NILAYAM India

Dynamic Programming: Decode-Ways, Find-the-Derangement-of-An-Array,String-Compression, Count-All-Possible-Routes,Number-of-Sets-of-K-Non-Overlapping-Line-Segments, Number-of-Ways-to-Form-a-Target-String-Given-a-Dictionary, Count-Ways-to-Distribute-Candies

References:

1. Lab Manual specific to the course compiled by DMACS will be provided
2. Trending problems suggested by Industrial Experts inline with the syllabus
3. Social platforms such as leetcode.com, hackerrank.com etc.

M.TECH. (COMPUTER SCIENCE)
LIST OF ELECTIVE COURSES (3CREDITS)

STREAM 1: INTELLIGENT SYSTEMS AND KNOWLEDGE ENGINEERING

ISKE 1 ARTIFICIAL INTELLIGENCE

(3 Credits) (42 Periods)

Course Objectives:

The Course will introduce students to the mapping of Human Intelligence and Behaviour onto Digital Systems - Computers that Functions as Intelligent Systems and to familiarize students with Concepts, Agents - Modeling, Intelligent Search, Problem Solving, Logic Reasoning, Knowledge Representation, Different Types of Learning.

Course Outcome: At the completion of the course the student will be able to:

- Know the theory and concepts behind Artificial Intelligence.
- Understand and appreciate the concepts such as Agents, State Space Search, Search Techniques, Ontological Engineering, Inductive learning etc.
- Understand the application of Propositional Logic and First Order Predicate Logic, for Reasoning Methods and Reasoning Process.
- Understand the concept of Learning and Different Types of Learning Models.

Course Syllabus"

Unit 1:

Introduction – what is AI? – Intelligent agents, environments – Solving problems by searching: problem solving agents –Example problems – Uninformed search strategies – Informed search and exploration: Informed search strategies –Heuristic functions– Local search algorithms – Optimization problems.

(10 periods)

Unit 2:

Logical Agents: Knowledge Based Agents – Logic – Propositional logic – Reasoning patterns – Propositional Inference – Agents based on propositional logic – First Order Logic :Representation – Using FOL – Knowledge Engineering – Inference in FOL: Unification And Lifting – Forward Chaining – Backward Chaining – Resolution – Examples.

(14 periods)

Unit 3:

Knowledge Representation: Ontological Engineering – Categories and objects – Actions situations and events – Mental events and mental objects – Reasoning Systems – Truth maintenance systems

(9 periods)

Unit 4:

Learning from Observations: Forms of learning – Inductive learning – Learning Decision trees - Knowledge in Learning – Knowledge in Learning – Explanation based learning – Learning using relevance information (9 periods)

Total (42 Periods)

Reference Text:

1.Stuart J. Russel and Peter, Norvig, *Artificial Intelligence – A Modern Approach,* Prentice Hall, Pearson Education, 2003.

[Chapters & Sections : 1: 1.1 ; 2: 2.1 to 2.4 ; 3:3.1 to 3.6; 4: 4.1 to 4.3 ; 7: 7.1 to 7.7; 8 : 8.1 to 8.4; 9: 9.1 to 9.5 ; 10: 10.1 to 10.8; 18: 18.1 to 18.3 ; 19: 19.1 to 19.4]

Suggested Readings:

1. George F. Luger and William A. Stubblefield, *Artificial Intelligence, Structures and Strategies for Complex Problem Solving,* The Benjamin / Cummings Publishing Co, 1993.
2. Amit Konar, *Artificial Intelligence and Soft Computing,* CRC Press, 2000.

ISKE 1(P) Practicals: ARTIFICIAL INTELLIGENCE (1 Credit) (28 periods)

Course Objectives:

The course introduces students to the practical aspects of the methods learnt in the theory course. The students will be made to implement some algorithms pertaining to important aspects of the course such as Search Techniques, Reasoning Techniques and Optimization Techniques.

Course Outcome: At the completion of the Course the student will be able to:

- Know the practical aspects of the theory and concepts learnt
- Understand the principles of Propositional and First Order Predicate Logic, Applications reasoning Methods and Reasoning Process through examples and simulations
- Creation of Knowledge Base by implementing Knowledge Representation Techniques such as Semantic Nets, Frames and Ontology Inject skills into a learner.

Syllabus:

- Informed Search Techniques using Heuristics and their Implementation
  - i) Steepest Hill Climbing Programming
  - ii) A Star Algorithm Programming to find shortest path / optimal path
  - iii) Genetic Algorithm Programming and Optimization
- Forward Chaining Programming
- Backward Chaining Programming
- Reasoning by Resolution

Reference Text:

1. Stuart J. Russel and Peter Norvig, *Artificial Intelligence – A Modern Approach*, Prentice Hall, Pearson Education, 2003.

Suggested Readings:

1. George F. Luger and William A. Stubblefield, *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*, The Benjamin / Cummings Publishing Co, 1993.

2. Amit Konar, *Artificial Intelligence and Soft Computing*, CRC Press, 2000.

***

ISKE 2 NATURAL LANGUAGE PROCESSING (3 Credits) (42 Periods)

Course Objectives:

The course introduces students to the principles, concepts and theory behind language processing. There is an introduction to languages grammar followed by transformational grammars of natural language. Transition networks and two level processing of moving from grammar to acceptor. The course will enable a student to learn and work on techniques and models to process sentences formed out of a natural language.

Course Outcome: At the completion of the course the student:

- will Know the basic concepts and theory of languages and grammar
- is expected to understand the syntax and semantics of the sentences formed out of a natural language
- will be able to understand transformational grammars of natural languages
- will know the application of Transition Networks from Grammar to Acceptor
- will know the two level processing systems namely RTNs and ATNs

Syllabus:

Introduction to Languages and Grammars	(4 periods)
Transformational Grammars of Natural Language	(6 periods)
Two-Level Representation	(7 periods)
Transition Networks - From Grammar to Acceptor	(9 periods)
Two Level Processing Systems RTN's and ATN's	(8 periods)
Issues and Applications	(8 periods)
Total	**(42 hours)**

Reference Text:

1. Gilbert K. Krulee, *Computer Processing of Natural Language*, Prentice Hall 1991.

ISKE 2(P) Practicals: NATURAL LANGUAGE PROCESSING (1 Credit)

(28 periods)

Course Objectives:

This lab based course introduces students to the practical aspects of Natural Language Processing. The subject encompasses and covers Parsing, Sentence Analysis, Application of Grammar and methods for semantic analysis of sentences. Other Grammars RTNs and ATNs are included for simulations. The course will enable a learner to take up studies in Document Processing.

Course Outcome: At the completion of the Course the student will be able to:

- Know the practical aspects of the theory and concepts of Natural Language Processing (NLP).

Recommended Assignments:

- Sentence Parser and application of Grammar through simulations
- Semantic Analysis
- Total Sentence Analysis for document processing
- Transformational Grammar, Transition Networks for simulations
- RTN and ATN networks.

Reference Text:

1. Gilbert K. Krulee, *Computer Processing of Natural Language*, Prentice Hall, 1991.

***

ISKE 3 Machine Learning (2 Credits) (28 Periods)

Prerequisite: First level course in Multivariable calculus, linear algebra, probability and statistics

Course objectives:

To provide students with an in-depth introduction to two main areas of Machine Learning: supervised and unsupervised. The contents will cover some of the main models and algorithms for regression, classification and clustering. Topics will include linear and logistic regression, MLE, probabilistic (Bayesian) inference, clustering and model selection.

Course outcomes:

- Develop an appreciation for what is involved in learning from data.
- Understand some of the important learning algorithms pertaining to classification and regression.
- Appreciate as to how probabilistic framework is applied for building various models of learning
- Understand how to apply a variety of learning models for the tasks of classification, regression and clustering.

Unit I : What is Machine Learning? Why use Machine Learning? Types of Machine Learning- Main challenges of Machine Learning (1 Period)

Unit 2: Classification and Regression – Binary classification– Multiclass and multilabel classification – Performance measures - Error analysis - VC dimension - PAC learning (6 Periods)

Unit 3: Bayesian Decision Theory and Parametric Methods -Discriminant functions - MLE estimation - Bias and Variance Dilemma - Model selection procedures (4 periods)

Unit 4: Training models – Linear regression – Regularization – Logistic Regression (2 Periods)

Unit 5: Support vector machines – Linear SVM – Nonlinear SVM – Kernel SVM - SVM regression (3 Periods)

Unit 6: Decision Trees – Information Gain – Gini impurity measure – CART algorithm – Regression (3 Periods)

Unit 7: Ensemble Learning – Random forests – Bagging and pasting – Boosting - Stacking (3 Periods)

Unit 8: Dimensionality Reduction – PCA – tSNE – kernel PCA - LLE (3 periods)

Unit 9: Clustering - Mixture densities - k means clustering - EM algorithm - Hierarchical clustering (3 periods)

Total 28 Periods

References:

1. Aurelien Geron, O'Reilly Media, *Hands-on Machine Learning with Scikit-ILearn and TensorFlow, Firstedition, 2nd release, 2017, ISBN: 978-1-491-96229-9 2017*
2. Ethem ALPAYDIN, *Introduction to Machine Learning*, The MIT Press, February 2010, ISBN-10: 0 262 01243 X, ISBN 13: 978 0 262 01243 0
3. http://cs229.stanford.edu/
4. https://work.caltech.edu/telecourse.html
5. Tom Mitchell, *Machine Learning*, McGraw Hill (Oct 1997).

***

ISKE 3(P) Practicals: MACHINE LEARNING (2 Credits) (56 periods)

Course objectives: To provide the student with necessary skill set for implementing some important learning algorithms on realistic data. To make the student appreciate concepts such as overfitting, training data, test data, model validation etc.

Course outcomes: At the end of this lab course, a student will be able to

- Do basic cleaning of the data to suit the implementation
- Code the specific algorithm in a language such as python
- Report the performance of the implemented code through necessary graphs/tables
- Argue about the performance and say why or why not the algorithm is behaving in a particular manner Recommended assignments:

1. Implement learning algorithms for classification
2. Implement learning algorithms for regression
3. Implement learning algorithms for clustering

Datasets may be taken from standard websites which pertain to realistic scenarios

References:

1. Aurelien Geron, O'Reilly Media, *Hands-on Machine Learning with Scikit-ILearn and TensorFlow, First edition, 2nd release, 2017, ISBN: 978-1-491-96229-9 2017*

2. Ethem ALPAYDIN, *Introduction to Machine Learning*, The MIT Press, February 2010, ISBN-10: 0-262-01243-X, ISBN-13: 978-0-262-01243-0

***

ISKE 4 Mining of Big Data Sets (2 Credits) (28 Periods)

Course objectives:
The course will discuss data mining and machine learning algorithms for analyzing very large amounts of data. The emphasis will be on Map Reduce and Spark as tools for creating parallel algorithms that can process very large amounts of data. Topics include: Frequent itemsets and Association rules, Near Neighbor Search in High Dimensional Data, Locality Sensitive Hashing (LSH), Dimensionality reduction, Recommendation Systems, Clustering, Link Analysis, Data streams, Mining.

Course outcomes: At the end of the course, a student must be able to

- Understand the significance of scaling up an algorithm for huge datasets
- Implement a few algorithms in Map reduce paradigm
- Understand the importance and implement randomized & hashing algorithms that scale up well with data size without sacrificing much on accuracy

Course Syllabus:

UNIT-I (4 Periods)

Data Mining: Bonferroni's Principle - Hash Functions - Power Laws.
MapReduce and the New Software Stack - Distributed File Systems - Physical Organization of Compute Nodes - Large-Scale File-System Organization – MapReduce - Extensions to MapReduce -Workflow Systems - Recursive Extensions to MapReduce - Pregel

The Communication Cost Model- Communication-Cost for Task Networks - Complexity Theory for MapReduce - Reducer Size and Replication Rate - A Graph Model for MapReduce Problems - Case Study: Matrix Multiplication

UNIT-II (6 Periods)

Finding Similar Items - Applications of Near-Neighbor Search - Jaccard Similarity of Sets-Collaborative Filtering as a Similar-Sets Problem - Shingling of Documents - k-Shingles - Choosing the Shingle - Hashing Shingles - Shingles Built from Words - Similarity-Preserving Summaries of Sets - Matrix Representation of Sets - Minhashing - Locality-Sensitive Hashing for Documents - LSH for Minhash Signatures - Analysis of the Banding Technique - Combining the Techniques - Distance Measures - The Theory of Locality-Sensitive Functions - Applications of Locality-Sensitive Hashing - Methods for High Degrees of Similarity - Finding Identical Items

UNIT-III (6 Periods)

Mining Data Streams - The Stream Data - Sampling Data in a Stream - A Motivating Example - Filtering Streams - The Bloom Filter - Counting Distinct Elements in a Stream - Estimating Moments - Counting Ones in a Window - Query Answering in the DGIM Algorithm – Decaying Windows

Link Analysis - PageRank - Early Search Engines and Term Spam - Efficient Computation of PageRank - PageRank Iteration Using MapReduce - Other Efficient Approaches to PageRank Iteration - Topic-Sensitive PageRank - Link Spam - Architecture of a Spam Farm - Analysis of a Spam Farm - Combating Link Spam – Trust Rank - Spam Mass

UNIT-IV (6 Periods)

Frequent Itemsets - Association Rules - Finding Association Rules with High Confidence - Use of Main Memory for Itemset Counting - Monotonicity of Itemsets - Handling Larger Datasets in Main Memory - The Multistage Algorithm - The Multihash - The SON Algorithm and MapReduce - Counting Frequent Items in a Stream

UNIT- V (6 Periods)

Clustering - Introduction to Clustering Techniques - Clustering Strategies - The Curse of Dimensionality - Hierarchical Clustering - Hierarchical Clustering in Non-Euclidean Spaces - K-means Algorithms - The Algorithm of Bradley, Fayyad, and Reina - The CURE Algorithm - Clustering for Streams and Parallelism

Reference Text: Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Mining of Massive Datasets, Cambridge University Press, 2nd Edition(2010)

The text book will be supplemented with research papers and assignments & Projects designed for the course by the instructor.

ISKE 4(P) Practicals: Mining of Big Data Sets (2 Credits) (56 periods)

Course objectives: To provide the student with necessary skill set for implementing some important learning algorithms on large data. To make the student appreciate concepts such as scaling in computation with data size, randomization and hashing strategies for getting required performance and accuracies etc.

Course outcomes: At the end of this lab course, a student will be able to

- Model some big data related algorithms for Map-Reduce paradigm
- Implement specific algorithms on Map-Reduce platforms such as SPARK
- Study the performance of the implemented code through experimentation on standard datasets
- Analyse about the performance and say why or why not the algorithm is behaving in a particular manner

Course Syllabus:

Map Reduce, Finding Similar Items, Mining Data Streams, Link Analysis,
Frequent Item-sets, Clustering, Recommendation System related Algorithms/Exercises from the syllabus will be implemented in Lab. Students may be encouraged to do a mini project.

Reference Text: Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, *Mining of Massive Datasets*, Cambridge University Press, 2nd Edition, 2010.

***

ISKE 5 Deep Learning (2 Credits) (28 periods)

Course Objectives:

- To Introduce deep learning (DL) algorithms including convolutional neural networks (CNN), recurrent neural networks (RNN) and its variants viz. LSTM and GRU
- To train on how to fine tune hyper parameters of DL algorithms
- To impart concepts that help identify suitable applications for CNN, RNN, LSTM and GRU and study them

Course Outcomes:

A student at the end of course should be able to

- Decide if DL is suitable for a given problem
- Choose appropriate DL algorithm to solve the problem with appropriate hyper parameter setting
- Feel comfortable to read and understand DL articles from reputed conferences, journals including NIPS, CVPR, ICCV, ICML, PAMI etc.

Course Syllabus:

Unit I : Introduction - What is Deep Learning? – Perceptron and Multi-layer Perceptron – Hebbian Learning - Neural net as an Approximator - Training a neural network - Perceptron learning rule - Empirical Risk Minimization - Optimization by gradient descent (4 Periods)

Unit 2: Back Propagation - Calculus of Back Propagation (4 Periods)

Unit 3: Convergence in Neural networks - Rates of Convergence – Loss Surfaces – Learning rate and Data normalization – RMSProp, Adagrad and Momentum (4 Periods)

Unit 4: Stochatic Gradient Descent - Acceleration – Overfitting and Regularization – Choosing a Divergence Loss Function – Dropout – Batch Normalization (4 Periods)

Unit 5: Convolutional Neural Networks (CNN) - Weights as Templates – Translation Invariance – Training with shared parameters – Arriving at the convolutional model - Mathematical details of CNN – Alexnet – Inception – VGG - Transfer Learning (6 Periods)

Sri Sathya Sai Institute of Higher Learning (Deemed to be University) PRASANTHI NILAYAM India

Unit 6: Recurrent Neural Networks (RNNs) - Modeling sequences - Back propagation through time - Bidirectional RNNs - Exploding/vanishing gradients - Long Short-Term Memory Units (LSTMs) (6 Periods)

Total 28 Periods

Reference Text:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning, Online book*, 2017
2. Michael Nielsen, *Neural Networks and Deep Learning, Online book*, 2016

Suggested Readings:

1. http://cs231n.github.io/
2. https://www.coursera.org/specializations/deep-learning
3. https://cilvr.cs.nyu.edu/lib/exe/fetch.php?media=deeplearning:2015:a4.pdf
4. http://deeplearning.cs.cmu.edu/

ISKE 5(P) Practicals: Deep Learning (2 Credits) (56 periods)

Course Objectives:

- To Introduce PyTorch / Tensor Flow as a deep learning framework
- Train on Implementing various DL algorithms studied in theory
- To become good in analyzing the influence of hyper parameters
- To have hands on through a mini project

Course Outcomes: A student should be

- Able to understand and analyze existing codes in PyTorch/Tensor Flow from Github
- Able to implement proposed algorithms on reputed frameworks or design a his/her own architecture to suite the need

Lab exercises can be based on the recommended list given below. Lab exercises could be implemented in Python 3 and Tensor Flow/PyTorch.

Lab 1: Implement a shallow network in Python for a binary classification problem

Lab 2: Make the network of Lab1 deeper and compare the performance of two networks on various aspects

Lab 3: Build a fully connected deep network to classify Cifar10 dataset

Lab 4: Analyze convergence of Lab3 network and improvise using various methods studied to accelerate convergence

Lab 5: Analyze over-fittingness of lab3 network and improvise using regularization methods studied

Lab 6: Build a cnn using basic python and numpy to classify Cifar10 data

Lab 7: Play with Pytorch / Tensorflow by going through online tutorials

Lab 8: Implement lab 7 network in Pytorch/Tensorflow

Lab 9: Train a RNN language model to do word-level and character level prediction

Reference Text:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning, Online book,* 2017
2. Michael Nielsen, *Neural Networks and Deep Learning, Online book,* 2016

Suggested Readings:

1. http://cs231n.github.io/
2. https://www.coursera.org/specializations/deep-learning
3. https://cilvr.cs.nyu.edu/lib/exe/fetch.php?media=deeplearning:2015:a4.pdf
4. http://deeplearning.cs.cmu.edu/

***

ISKE 6 Fundamentals of Blockchain Technologies and Applications

(3 Credits) (42 periods)

Course Objective:

This course covers different aspects of Blockchain. Role of Distributed systems, including P2P networks, distributed consensus are discussed. Bitcoin, Ethereum and Hyperledger are introduced and the main components are discussed. Blockchain technologies, cryptographic techniques (secure hashing, encryption, decryption, digital signatures), privacy and anonymity, mining and mining puzzles, wallets, smart contracts are discussed along with some aspects of their applications.

Course Outcome: By the end of the course, the students will be able to

- Discern the infrastructural needs of Blockchain
- Recognize the complexities involved in managing Blockchain platforms
- Understand the intricacies in peer-to-peer consensus systems
- Identify the role of cryptographic techniques in Blockchain
- Recognize the advantages of smart contracts and their applications
- Discern the concept of mining cryptocurrencies

- Discern the need of different types of Blockchain for different applications
- Understand the importance of Blockchains and their growing demand in industry
- Identify the role of Blockchain data structures in ensuring data permanency
- Understand characteristics of electronic cash/currencies
- Recognize the concepts and techniques used in Ethereum
- Identify the concepts and techniques used in Bitcoin
- Identify the concepts and techniques used in Hyper Ledger

Unit#	Description	#of periods	Reading material
Unit 1	Distributed systems - Clock Synchronization, Multicast Communication, Fault tolerance, Replication, Ordering, Consistency, Consensus in synchronous and Asynchronous systems, peer-to-peer systems, availability, reliability and quality of service. Security and Trust.	2	Reference 1
Unit 2	Cryptography --- symmetric and asymmetric key encryption algorithms, the secure hashing algorithms, digital signatures and message authentication algorithms.	2	Reference 1
Unit 3	Data structures in Blockchains Merkle Tree, Trie, and PATRICIA Tree	2	Textbooks 1&2
Unit 4	Blockchain Foundations: decentralized architecture and the cryptographic techniques employed, provenance, functionality and trust. Benefits of blockchains like cost savings, transparency, anonymity, lower risk, efficiency. Illustrative applications.	2	Textbook 1
Unit 5	Consensus Mechanisms in Blockchains: proof-of-work, proof-of-stake, Paxos, Byzantine agreement, Practical Byzantine Fault Tolerance, RBFT and Federated agreement.	4	Textbook1, references 3 &4

Unit 6	Cryptocurrencies – Basics. advantages and disadvantages of E-cash systems and their desirable characteristics. Untraceable electronic cash, traceable anonymous cash, and untraceable electronic cash protocols. Three real E-cash products of the past---First Virtual, Digicash, and Mondex	2	Textbooks 1&2, Reference 5
Unit 7	Bitcoin : History, mechanics, ways to store and exchange bitcoins, mining, anonymity features, and current applications.	4	Textbook 1&2
Unit 8	Ethereum : technical foundations and the implementation details of Ethereum. It discussed the concepts of state, transactions, and blocks, the consensus algorithm employed, the data structures used, the proof-of-work algorithm, contract creation, and Ethereum Virtual Machine.	4	Textbook 2, Reference 6
Unit 9	Smart contracts: smart contracts in the context of Ethereum. Different features of smart contracts, justification for smart contracts, example applications, building blocks for smart contracts	2	Reference 7
Unit 10	Hyperledger : Applications in Banking, Financial Services, Healthcare, Supply Chain Management. Hyperledger Frameworks: Burrow, Fabric, Indy, Iroha, Sawtooth. Hyperledger Tools: Caliper, Cello, Composer, Explorer, Quilt	4	Reference 8, Textbook 2
Unit 11	IOT & Blockchain : IoT security requirements with a focus on intentional risk, IAM, IOTA IoTeX.	2	Textbook 2, Reference 9
Unit 12	Privacy & Anonymity: Privacy Tiers, Privacy techniques, Attacks, Commitments, Pedersen, HashBased, Zero Knowledge proof, Bulletproofs,	2	Reference 10
Unit 13	Applications of Blockchains: Applications in a variety of application domains including supply chains, land registry, border control, agriculture, healthcare, energy, etc. In addition, applications in governments including elections, marriage registrations, passport issuances, banking, etc.	8	Reference 11, Textbook 1

Unit 14	Future of Blockchains: AI & Blockchain	2	Reference 12
	TOTAL	42	

Text Books, Reading materials, References:

Textbook 1: A. Narayan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, "Bitcoin and Cryptocurrency Technologies," Princeton University Press, 2016. http://press.princeton.edu/titles/10908.html

Textbook 2: Stijn Van Hijfte, "Blockchain Platforms" Morgan & Claypool, 2020.

Reference 1: G. Colouris, J. Dollimore, T. Kindberg, and G. Blair, "Distributed Systems: Concepts and Design," 5th Edition, Pearson, 2012.

Reference 2: https://www.youtube.com/watch?v=fw3WkySh_Ho

Reference 3: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

Reference 4: https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf

Reference 5: David Chaum, Amos Fiat, and Moni Naor, "Untraceable electronic cash," in Adavnces in Cryptology – (CRYPTO'88), LNCS 403, pp. 319-327, 1990, Springer-Verlag.

Reference 6: https://github.com/ethereumbook/ethereumbook/blob/develop/what-is.asciidoc

Reference 7: Kevin Delmolino, Mitchell Arnett, Ahmed E Kosba,Andrew Miller, and Elaine Shi. 2015. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. IACR Cryptology ePrint Archive 2015 (2015), 460.

Reference 8: "An introduction to Hyperledger," HL_Whitepaer_IntroductiontoHyperledger.pdf

Reference 9: A. Partida, R. Criado, and M. Romance, "Identity and Access Management Resilience against Intentional Risk for Blockchain-Based IOT Platforms," electronics-10-00378-v2 (1).pdf

Reference 10: N. Alsalami and B. Zhang, "SoK: A Systematic Study of Anonymity in Cryptocurrencies," 266984898.pdf (core.ac.uk)

Reference 11: JS Arun, J. Cuomo, and N. Gaur, "Blockchain for business," Addison Wesley, 2019. IBM Blockchain Platform - IBM Blockchain | IBM

Reference 12: S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, PK Singh, and W-C. Hong, "Machine Learning Adoption in Blockchain-Based Smart Applications: The Challenges, and a Way Forward," IEEE Explore, 2020.

ISKE 6(P) Practicals: Fundamentals of Blockchain Technologies and Applications **(1 Credit) (28 periods)**

Course Objective: This course is to help gain hands-on experience on the concepts delivered in the associated theory course. Experimental study to comprehend performance bottlenecks in Applications developed on a Blockhead Platform. Study possible ways to alleviate the bottlenecks.

Course Outcome: By the end of the course, the students will be able to

- Understand Solidity language, Implement a smart contract
- Develop simple applications on Hyperledger fabric using ChainCode
- Measure experimentally throughput and latency in transaction processing
- Recognize the concepts and techniques used in BigChainDB & HBasechainDB

Syllabus Contents:

Unit No	Contents	Periods
Unit I	Ethereum: Solidity language to develop applications in Ethereum and discover limitations due to high latency and low throughput	4
Unit II	Hyperledger Fabric: Chain Code and study Consensus and Fault Tolerance	4
Unit III	BigChainDB: Fundamentals to appreciate and comprehend the design issues for gaining High throughput of Transaction and Blockchain for BigData environment.	4
Unit-IV	Smart Contract and Applications	4
Unit-V	Mini Project	12
	Total	28

Reference: 1. All the references used for the Associated Theory course.

2. Whitepapers, Manuals and Technical papers as prescribed by the Instructor.

***

Applicable from the academic year 2022-23 onwards

Sri Sathya Sai Institute of Higher Learning (Deemed to be University) PRASANTHI NILAYAM India

STREAM II: ADVANCED COMPUTER NETWORKS

ACN 1 WIRELESS AND MOBILE NETWORKS

(3 Credits) (42 Periods)

Course objectives:

The objectives are to impart fundamental knowledge of wireless network architectures, algorithms, protocols and applications, with special emphasis on local area networks, wide area networks, mobile ad hoc networks and wireless sensor networks.

Course outcomes:

Upon completing the course, the student will be able to:

- Understand how wireless local area networks operate and understand the corresponding design principles and protocols
- Understand the protocols and operation of wireless wide area networks, such as cellular networks
- Understand the challenges, protocols and mechanisms used for supporting medium access, routing and quality of service in mobile ad hoc networks and wireless sensor networks.

Course Syllabus:

Unit 1: (6 periods)
Wireless Local Area Networks - IEEE 802.11 family and related protocols; (Chapters 1,2 from key text)

Unit 2: (12 periods)
Wireless Wide Area Networks - 3G/LTE/WiMAX;
(Chapters 3 from key text, Chapters 1 and 2 from Ref Book 4by Martin Sauter)

Unit 3: (15 periods)
Mobile Ad Hoc Networks - Medium access control and Routing protocols;
(Chapters 5 to 7 from key text)

Unit 4: (9 periods)
Quality of service in Mobile Ad Hoc Networks, Wireless Sensor Networks - IEEE 802.15.4.
(Chapters 10.1-10.5, 12 from key text)

Total 42 Periods

Reference Text Book:

1. C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc wireless networks: architectures and protocols*, Prentice Hall PTR, 2004.

Suggested Readings:

2. Jochen H. Schiller, *Mobile Communications*, 2nd edition, Addison-Wesley, 2003, ISBN 0-321-12381-6

3. William Stallings, *Wireless Communications & Networks* (2nd Edition), 2004.

4. Martin Sauter, *Beyond 3G - Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0* Wiley; 1 edition (February 17, 2009) ISBN-10: 0470751886.

5. Kaveh Pahlavan, *Principles of Wireless Networks: A Unified Approach*, 2nd Edition, 2012 (Expected), ISBN-13: 978-0470697085, Wiley.

6. Erik Dahlman, Stefan Parkvall, Johan Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*, ISBN-10: 012385489X, ISBN-13: 978-0123854896, Academic Press, 2011.

7. Jonathan Loo, Jaime Lloret Mauri, *Jesus Hamilton Ortiz, Mobile Ad Hoc Networks: Current Status and Future Trends*, CRC Press, 2011, Edited Book.

8. Shih-Lin Wu, Yu-Chee Tseng, *Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks*, Auerbach Publications, 2007, Edited Book.

9. Rajeev Shorey, A. Ananda, Mun Choon Chan, Wei Tsang Ooi, Mobile, *Wireless, and Sensor Networks: Technology, Applications, and Future Directions*, Wiley-IEEE Press, 2006, ISBN-13: 978-0471718161, Edited Book.

ACN 1(P) Practicals: WIRELESS AND MOBILE NETWORKS (1 Credit) (28 periods)

Course objectives: This course provides practical training to supplement the objectives theory course

Course outcomes: At the end of the course the student will have the practical knowledge of some of the theoretical concepts taught in the theory course.

Syllabus: Some simulation studies based on concepts/technology topics from the theory course can be undertaken on standard simulators such as OMNET++, NS(3) or Wireshark.

Reference Text Book:

1. C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc wireless networks: architectures and protocols*, Prentice Hall PTR, 2004.

2. http://ns3simulation.com/list-of-network-simulators/

***

STREAM III: HUMAN COMPUTER INTERACTION

HCI 1 DIGITAL IMAGE PROCESSING (3 Credits) (42 Periods)

Course Objectives: This course will provide an introduction to the fundamental concepts of digital image processing. The course will focus on image fundamentals, pre processing and segmentation.

Course Outcomes: Upon completion of the course, the students will be

- Having a broad understanding of fundamentals of digital image processing and will be able to appreciate the challenges in computer vision domain.
- Familiarity with various digital image properties and ability to pre-process image for a particular application.
- Understand the utility of spatial and frequency domain analysis
- Understand basic image segmentation techniques and their application scenarios

Unit 1: Introduction - What is digital image processing, examples of fields that use digital image processing, fundamental steps in digital image processing, components of an image processing system (4 periods)

Unit 2: Digital image fundamentals - elements of visual perception, light and electromagnetic spectrum, image sensing and acquisition, image sampling and quantization, basic relationships between pixels, introduction to mathematical tools used in digital image processing (6 periods)

Unit 3: Intensity transformations and spatial filtering – Basic intensity transformation functions, histogram processing, fundamentals of spatial filtering, smoothing spatial filters, sharpening spatial filters (10 periods)

Unit 4: Filtering in frequency domain – basics of filtering in frequency domain, image smoothing using lowpass frequency domain filters, image sharpening using highpass filters, selective filtering (10 periods)

Unit 5: Image segmentation - fundamentals, point, line and edge detection, thresholding, segmentation by region growing and by region splitting and merging (12 periods)

Total :42 periods

TEXTBOOK

1. Rafael C. Gonzalez & Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2018.

2. Relevant research papers selected for the course by the instructor

REFERENCE BOOKS

1. Maria Petrou and Costas Petrou, Image Processing – The fundamentals by, 2nd Edition, John Wiley and Sons, 2010.

2. Anil K. Jain,Fundamentals of Digital Image Processing, Eastern Economy Edition, Prentice Hall, India, 1997.

***

HCI 1(P) Practicals: DIGITAL IMAGE PROCESSING (1 Credit) (28 Periods)

Course Objectives: In this course, the student will gain hands on training on various themes discussed in the (HCI-1) digital image processing lectures. This course will clarify the concepts and principles and help enhance the theory-to-code skill.

Course Outcomes: By the end of this course, the students will be introduced to the basic know-how to convert relevant image processing mathematics to code.

Syllabus:

- Read, write and manipulate digital images.
- write a Programme to implement basic image processing algorithms related to enhancement, restoration, filtering and segmentation.

Reference Textbook(s):

Rafael. C. Gonzalez & Richard E. Woods, *Digital Image Processing*, 3rd Edition, Pearson Education, 2002.

* * *

HCI 2 MEDICAL IMAGE PROCESSING (3 Credits) (42 Periods)

Course Objectives:

This course will provide an introduction to various imaging modalities such as CT, Ultrasound, X-Ray, MRI in the medical imaging field. The mathematics behind the subject (transform techniques) are also covered for providing an in-depth understanding of the imaging techniques. The course also achieves a node acquaintance with various methods and aspects of the medical imaging domain.

Course Outcomes: Upon completion of the course, the students will be

- Having a broad understanding of fundamentals of medical imaging modalities such as X-Ray, CT, Ultrasound and MRI.
- Able to apply basic mathematical tools such as transform techniques on the images
- Familiar with the modeling and analysis of reconstruction, relaxation and contrast enhancement mechanisms.
- Visualize medical data in an appropriate software

Course Syllabus:

Historical perspective -Generic Principles – modality – contrast – SNR – resolution – toxicity - Measurements and Modeling: Review of Linear Systems and Models – Basic Model for Tomography - Sampling - Fourier and Hankel transforms - k-space -

XRay projection radiography – Reconstruction in X-Ray Tomography - Computerized Tomography - acquisition and reconstruction methods - relaxation and contrast mechanisms – applications - Nuclear medicine - radionuclides, PET, SPECT imaging – Applications of Probability : PET.

Ultrasound Imaging - echo equation - beam forming - Medical Image Processing - physics of Magnetic resonance imaging - MRI reconstruction, functional MRI.
Fuzzy and Neuro Fuzzy Systems: Medical Image Analysis and Processing – Wavelets and Fuzzy gated SPECT Images of Ventricles.

Visualization of medical imaging data-segmentation applications.

Reference Books:

1. Albert Macovski, Medical Imaging Systems, Prentice Hall, 1983.

2. Joseph Hornak, The Basics of MRI, Online at http://www.cis.rit.edu/htbooks/mri

3. Charles L. Epstein, Introduction to Mathematics of Medical Imaging, Pearson Education, Prentice Hall, NJ, 2003.

4. H.N. Teodorescu, L.C. Jain, Abraham Kandel, Fuzzy and Neuro Fuzzy Systems in Medicine, Computational Intelligence, CRC Press, 1999.

Suggested Reading:

1. John L Semmlow, *Biosignal and Biomedical Image Processing: MATLAB Based Applications*, CRC Press, 2008.

2. Kavyan Najarian, *Biomedical Signal and Image Processing*, CRC Press, 2012.

3. Isaac Bankmem, *Handbook of Medical Imaging: Processing and Analysis*, Academic Press, 2000.

4. Anil. K. Jain, *Fundamentals of Digital Image Processing*, Eastern economy ed., Prentice Hall of India, 1997.

HCI 2(P) Practicals: MEDICAL IMAGE PROCESSING (1 Credit) (28 periods)

Course Objectives: This course familiarizes the student with various phases of handling medical images - image acquisition, preprocessing and enhancement, processing, segmentation and analysis. The software used is 3D Slicer.

Course Outcomes: Upon completion of the lab course, the students will be

- able to acquire medical images in 3D slicer and learn various aspects of manipulating the image
- preprocessing and processing the image
- visualize medical data in an appropriate software

Reference Books:

1. Albert Macovski, *Medical Imaging Systems*, Prentice Hall, 1983.

2. Joseph Hornak, *The Basics of MRI,* Online at http://www.cis.rit.edu/htbooks/mri

3. Charles L. Epstein, *Introduction to Mathematics of Medical Imaging,* Pearson Education, Prentice Hall, NJ, 2003.

4. H.N. Teodorescu, L.C. Jain, Abraham Kandel, *Fuzzy and Neuro Fuzzy Systems in Medicine,* Computational Intelligence, CRC Press, 1999.

* * *

HCI 3 COMPUTER VISION (3 Credits)(42 Periods)

Prerequisite: Basic course on Image processing.

Course Objectives: This course will provide an extension to the material provided in (HCI-I) Digital Image Processing course. Students will be provided additional knowledge on segmentation, introduced to feature extraction and pattern analysis.

Course Outcomes: Upon completion of the course, the students will be

• Having fair knowledge of the computer vision theory in terms of active contours, feature extraction and pattern classification.

• Able to understand commonly prevalent computer vision applications with an awareness of research challenges therein.

• Able to attempt solutions to common computer vision problems.

Syllabus:

Unit 1: Image segmentation – Image segmentation using snakes, segmentation using level sets (12 periods)

Unit 2: Feature extraction – Boundary preprocessing, boundary feature descriptors, region feature descriptors, whole image features, scale invariant feature transformation (12 periods)

Unit 3: Image pattern classification – patterns and pattern classes, pattern classification by prototype matching, optimum statistical classifiers, neural networks and deep learning, deep convolution neural networks (18 periods)

TEXTBOOK

1. Rafael C. Gonzalez & Richard E. Woods, Digital Image Processing 4th Edition, Pearson

Education, 2018.
2. Relevant research papers selected for the course by the instructor

REFERENCE BOOKS

1. Maria Petrou and Costas Petrou, Image Processing – The fundamentals by 2nd Edition, John Wiley and Sons, 2010

2. Anil K. Jain, Fundamentals of Digital Image Processing, Eastern Economy Edition, Prentice Hall, India, 1997.

***

HCI 3(P) PRACTICALS: COMPUTER VISION (1 Credit) (28 periods)

Course Objectives:

In this course, the student will implement and analyze chosen algorithms from computer vision literature to understand in depth the lectures of (HCI-3) Computer Vision course. The students will also analyze simple real life applications and their implementations demonstrating computer vision challenges.

Course Outcomes:

By the end of this course, the students will be

- Able to use OpenCV to implement and test computer vision algorithms
- Able to prototype image analysis and understand solutions for specific scenarios.

Syllabus:

Algorithms/Exercises from different units in the theory course can be taken up for lab exercise.

* * *

STREAM IV: THEORETICAL COMPUTER SCIENCE

TCS 1 ADVANCED ALGORITHMS

(3 Credits) (42 Periods)

Course Objectives:

The course handles problems that are NP-Complete and NP-Hard through approximation and randomization. Broadly speaking, the approximate algorithms only provide a suboptimal solution and take polynomial time in the input size. The course also provides tools for performing a probabilistic analysis of both approximate and randomized algorithms.

Course Outcomes:

At the completion of the course the student will be able to

- Know how to design an approximate algorithm by inventing the necessary heuristic.
- Know how to convert a LP problem solved using simplex method into an approximate version.
- Understand Matroid and design greedy algorithms.
- Design randomized algorithms.
- Design and analyze randomized data structures.

Unit 1: Preliminaries 7 (Periods)

Complexity classes-Lower bounding OPT – Well Characterized problems – MinMax relations - Chernoff bounds – The Minmax principle – Randomness and Non-Uniformity – Occupancy problems – Two point sampling – Stable marriage problem – Coupon Collector's problem

Unit 2: Approximate Algorithms 12 (Periods)

Matroid and greedy methods - Min cut algorithm – Las Vegas – Monte Carlo – Set Cover – Greedy algorithm – LP duality – Dual fitting – Rounding – Primal Dual Schema, Knapsack – pseudo-polynomial time algorithm – FPTAS

Unit 3: Randomized Algorithms 15 (Periods)

Probabilistic Recurrence - Randomized selection - Delaunay Triangulation- Minimum Spanning Trees - Counting Problems

Unit 4: Advanced data structures 8 (Periods)

Fundamental data structuring problem – Random Treaps - Skip lists – Hash tables with O(1) search time

Total 42 Periods

Reference Books:

1. Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms*, Cambridge University Press, 2013
   [Chapters : 1, 2, 3, 4 (only 4.1), 8, 9(only 9.6), 10(only 10.3)]

2. Vijay V. Vazirani, Springer, *Approximate Algorithm*, 2003
[Chapters: 1, 2, 8, 12, 13, 14, 15, 28]

3. Thomas H. Cormen, Ronald L. Rivest and Clifford, *Introduction to Algorithms*, 2nd edition, MIT Press and McGraw Hill, 2001.
[Chapter: 16]

TCS 1(P) Practicals: ADVANCED ALGORITHMS (1 Credit) (28 periods)

Course Objectives:
Algorithms/Exercises from different units in the syllabus will be implemented in the Lab. The language of choice is Python.

Course Outcomes: At the completion of the lab course the student will be able to

- Know how to design an approximate algorithm by inventing the necessary heuristic.
- Know how to convert a LP problem solved using simplex method into an approximate version.
- Understand Matroid and design greedy algorithms.
- Design randomized algorithms.
- Design and analyze randomized data structures.

Syllabus:

- write an approximate algorithm for a given NP-Complete problem.
- write an approximate version of the LPP.
- Code and test the performance of randomized algorithms
- Code and test the performance of randomized data structures.

Reference Books:

- Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms*, Cambridge University Press
- Vijay V. Vazirani, *Approximate Algorithm*; Springer
- Thomas H. Cormen, Ronald L.Rivest and Clifford, *Introduction to Algorithms*, 2nd edition, MIT Press and McGraw-Hill, 2001.

* * *

TCS 2 CRYPTOGRAPHY (3 Credits) (42 Periods)

Course Objectives:
To provide foundations in cryptography for those students interested in pursuing cryptography, data privacy and network security.

Course Outcome: The students will be able to
- Understand the basics of cryptography
- Appreciate the philosophy behind symmetric and asymmetric key cryptography.
- Know when to and how to use public key primitives.

Course Syllabus:

Unit 1 : Introduction (2 periods)
OSI Security Architecture – Security Attacks - Security services – Security Mechanisms – A Model for Network Security

Unit 2 : Classical Cryptography Techniques (3 periods)
Symmetric Cipher Model-Substitution Techniques – Transposition Techniques – Rotor Machines – Steganography

Unit 3 : Block Cipher and the Data Encryption Standards (4 periods)
Block Cipher Principles – The Data Encryption Standards – The Strength of DES – Differential and Linear Cryptanalysis – Block Cipher Design Principles

Unit 4 : Finite Fields (3 periods)
Groups, Rings and Fields - Modular Arithmetic – The Euclidean Algorithm – Finite Fields of the Form GF(p) – Polynomial Arithmetic – Finite Fields of the Form GF(2^n)

Unit 5: Advanced Encryption Standard (4 periods)
Evaluation Criteria for AES – The AES Cipher

Unit 6 : More on Symmetric Ciphers (3 periods)
Multiple Encryption and Triple DES – Block Cipher Modes of Operation – Stream Ciphers and RC4

Unit 7 : Confidentiality Using Symmetric Encryption (4 periods)
Placement of encryption functions – Traffic Confidentiality – Key Distribution – random number Generation

Unit 8 : Introduction to Number Theory (3 periods)
Prime Number – Fermat's and Euler's Theorems – Testing for Primality – The Chinese Reminder Theorem – Discrete Logarithm Problem

Unit 9 : Public-Key Cryptographic and RSA (4 periods)

Principles of Public-Key Cryptosystems – The RSA algorithms - Key Management-Diffie-Hellman key Exchange - Elliptic Curve Architecture and Cryptography

Unit 10: Message Authentication and Hash Functions (4 periods)
Authentication requirements – Authentication functions – Message Authentication Codes – Hash functions – Security of Hash functions and MACs

Unit 11: Hash and MAC Algorithms (4 periods)
Secure Hash algorithm – Whirlpool – HMAC – CMAC

Unit 12: Digital Signatures and Authentication Protocols (4 periods)
Digital Signature – Authentication Protocols – Digital Signature Standards

Total **42 periods**

Reference Text:

1. William Stallings, *Cryptography and Network Security - Principles and Practices*, Prentice Hall of India, 4th Edn, 2003.
[Chapters : 1 to 13 (except Recommended readings and Appendices from all chapters)]

Suggested Readings:

1. Atul Kahate, *Cryptography and Network Security*, Tata McGraw -Hill, 2003.
2. Bruce Schneier, *Applied Cryptography*, John Wiley & Sons Inc, 2001.

***

TCS 2(P) Practicals: CRYPTOGRAPHY (1 Credit) (28 periods)

Course Objective: To provide hands-on sessions for some cryptography primitives and attacks.
Course Outcome: The students will be able to implement some cryptography primitives and attacks.

Suggested Assignments:

- Substitution and transposition ciphers.
- Block ciphers, modes of operation and stream ciphers.
- Hash functions and Message authentication codes.
- Public cryptographic primitives like RSA.
- Digital signatures.

Reference Text:

1. William Stallings, *Cryptography and Network Security - Principles and Practices*, Prentice Hall of India, 4th Edn, 2003.

* * *

TCS 3 DESIGN OF QUANTUM ALGORITHMS (3 Credits)

Course Objectives: This is an introductory course in Quantum Algorithms. After discussing required basic mathematical preliminaries and notation, the course discusses the basics ideas of quantum computing. All the key quantum algorithms will be explored and analyzed.

Course Outcomes:

(i) The student will be able to understand the basic structure of quantum algorithms, including the basic ideas of qubit, quantum representation of Boolean arguments and quantum circuit design.

(ii) The student can comprehend the mathematical aspects required for quantum computing and writing quantum algorithms.

(iii) The student will be able to grasp the nuances in designing quantum algorithms.

Content:

Unit 1: Preliminaries and Mathematical Basics: The Model, The Space and the States, The Operations, Input, Output, Asymptotic Notation, Hilbert Spaces, Tensor Products, Inner Products, Sum over Paths in Graphs 6 Periods

Unit 2: Quantum Bits: Feasible Boolean functions, Quantum Representation of Boolean Arguments, Quantum Feasibility 4 Periods

Unit 3: Special Matrices: Hadamard Matrices, Fourier Matrices, Reversible Computation, Permutation Matrices, Feasible Diagonal Matrices, Reflections 2 Periods

Unit 4: Tricks: Start Vectors, Controlling, Copying Base States, Copy-Uncompute Trick, Superposition Trick, Flipping a Switch, Measurement Tricks, Partial Transform. 4 Periods

Unit 5: Quantum Algorithms and Analysis: Phil's Algorithm, Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Simon's Algorithm, Shor's Algorithm, Factoring Integers, Gover's Algorithm 26 Periods

Total 42 periods

Key Text(s): Richard J. Lipton, Kenneth W. Regan, Quantum Algorithms via Linear Algebra - A Primer, MIT Press, 2014.

Coverage of Key Text(s): Chapters: 1 to 13

References: 1) https://qiskit.org/learn/intro-qc-qh

* * *

TCS 3(P) Practicals: DESIGN OF QUANTUM ALGORITHMS (1Credit) (28 periods)

Course Objectives: This course is the associated practical lab for the course Design of Quantum Algorithms. After familiarizing with the basic programming aspects associated with the quantum algorithm in python, the course discusses basic implementation of the operations on the quantum circuits. This lab course also familiarizes Shor's algorithm that can be used for solving the period-finding and integer factoring problems efficiently.

Course Outcomes:

(i) The student will have a practical knowledge of qubits, quantum circuits and measurements.

(ii) The student will be able to write and run quantum programs.

(iii) The student will be able to apply the key algorithms to solve basic computer science problems.

(iv) The student can design and implement certain computationally intractable problems using quantum programs.

Contents:

1 Python Programming Basics for Quantum Algorithms
2 Qubits, Quantum Circuits, Measurements,
3 Writing and Running Quantum Programs
4 Quantum Fourier Transform
5 Shor's Algorithm
6 Factoring Integers

References: 1) https://qiskit.org/learn/intro-qc-qh

***

STREAM V: COMPUTER SYSTEMS

CS 1 Advanced Programming in the UNIX Environment

(2 Credits) (28 Periods)

Course Objectives:

This course will introduce students to the basic and advanced features of the Unix Operating System. Features like, OS File system, Events generation and handling, Processes and their environment, Inter-process communication mechanisms, etc. Students with C background will code/Programme and familiarize with the programmatic perspective of the Unix Operating system.

They will perform some basic to advanced coding to customize and interact with the underlying OS. Students are introduced to the Unix OS library and utilize them to accomplish certain tasks with system level calls and features.

Course Outcomes:

At the completion of the course the student will be able to

- Utilize Unix OS system Library and the C Standard Library to access/manipulate the file system, internal data structures, etc.
- Understand and apply Multithreading concepts, Inter-process communication mechanisms like, Pipe, FIFO, Shared Memory, Semaphores.
- Understand and implement concepts to create processes, child processes, data structures used and invoke system libraries to manipulate processes.
- Code to generate the events, and handle different types of events generated by Kernel etc.

Course Syllabus:

Unit 1 : Introduction (4 Periods)

UNIX System Overview, UNIX Architecture Files and Directories, Input and Output, Error Handling, Signals, Time Values, System Calls and Library Functions

Unit 2 : File I/O (6 Periods)

File Descriptors, open and open at Functions, create Function, close Function, lseek Function, read Function, write Function, I/O Efficiency, File Sharing, dup and dup2 Functions sync, fsync, and fdatasync Functions, fcntl Function

Unit 3: Files and Directories (6 Periods)

stat, fstat, fstatat, and lstat Functions, File Types, File Access Permissions, Ownership of New Files and Directories, chmod, fchmod, and fchmodat Functions, chown, fchown, fchownat, and lchown Functions, link, linkat, unlink, unlinkat, and remove Functions, Creating and Reading Symbolic Links, Reading Directories, chdir, fchdir, and getcwd Functions, Device Special Files

Unit 4: Process Control (6 Periods)

fork Function, vfork Function, exit Functions, wait and waitpid Functions, waitid Function, wait3 and wait4 Functions, Race Conditions, exec Functions, Interpreter Files, system Function

Unit 5: Signals (6 Periods)

signal Function, Unreliable Signals, Interrupted System Calls, Reentrant Functions, SIGCLD Semantics, kill and raise Functions, alarm and pause Functions, sigprocmask Function, sigpending Function, sigaction Function, sigsetjmp and siglongjmp Functions, sigsuspend Function, sleep, nanosleep, and clock_nanosleep, sigqueue Function

Total (28 Periods)

Reference Text: Advanced Programming in the UNIX Environment Third Edition by W. Richard Stevens Stephen A. Rago, 1993.

CS 1(P) Practicals: Advanced Programming in the UNIX Environment

(2 Credits) (56 periods)

Course Objectives:
This course will introduce students to the hands-on training for the advanced features of the Unix Operating System.

Course Outcomes: At the completion of the course the student will have the practical knowledge of

- How to access/manipulate the file system, internal data structures.
- Implementing Multithreading, Inter-process communication mechanisms like Pipe, FIFO, Shared Memory, Semaphores.
- Implementing concepts to create processes, child processes, data structures used and invoke system libraries to manipulate process.
- Hands on exercises on Files and File system like creating a hole in a file, retrieving and changing File meta data and modify access permissions, accessing data structures of internal File system.
- Coding on creating processes and child processes and assigning tasks and communication between parent and children.

Reference Text:
W. Richard Stevens Stephen A. Rago, *Advanced Programming in the UNIX Environment*, Third Edition.

* * *

STREAM VI: MULTI-CORE AND PARALLEL COMPUTING

MPC 1 HIGH PERFORMANCE COMPUTING WITH ACCELERATORS

(2 Credits) (28 Periods)

Course Objectives:

To develop a student's ability in computing with accelerators to the level of designing high performance systems using Multiprocessor Design Techniques and Multiprocessor Software.

Course Outcomes:

By the end of the module the student should be able to:

- Apply the more advanced features of architectures in accelerators based systems design.

- Explain Multiprocessor Design and Interconnection Network for a High Performance Accelerators systems
- Integrate a hardware accelerator with a processor and design the necessary software and hardware communication infrastructure

Unit 1: Data Parallel Computing (6 periods)
Data parallelism, CUDA Kernel, Device Global memory, Function declaration, Kernel launch.

Unit 2: Scalabale Parallel execution (8 periods)
CUDA Thread organization, Synchronization, Scheduling and Latency Tolerance.
Memory and data Locality, Tiling for reduced memory Traffic, Boundary checks, Global Memory bandwidth, Warp, Dynamic Partitioning of Resources, Thread Granularity.

Unit 3: Parallel Algorithm Patterns (8 periods)
Convolution, Prefix Sum, Parallel Histogram, Sparse matrix Computation, Merge Sort, Graph Search.

Unit 4: CUDA Dynamic Parallelism and **Open ACC** (6 periods)

Case Study: Machine Learning.

Total: 28 Periods

Note: Depending on the circumstances, OpenCL could be used in place of CUDA (OpenCL is another language similar to CUDA which is gaining importance recently).

TEXT BOOKS

1. David B Kirk and Wen-mei W. Hwu, *Programming Massively Parallel processors, A hands on Approach*, Third Edition, Elsevier Morgan Kauffman, 2017.

REFERENCES:

1. Jason Sanders and Edward Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison Wesley Professional, 1st edition, 2010.
2. Wen-mei W. Hwu, *GPU Computing Gems*, Emerald Edition, Morgan Kauffman Publishers, 2011.
3. Benedict R. Gaster, et al., *Heterogeneous Computing with OpenCL Morgan Kaufmann*, 2012.

MPC 1(P) Practical: HIGH PERFORMANCE COMPUTING WITH ACCELERATORS

(2 Credits)(56 Periods)

Course Objectives:

To develop a student's ability in practical aspects of accelerator computing through simulation experiments

Course Outcomes:

By the end of the module the student should be able to demonstrate through simulations, the design principles of architectures in high performance accelerator systems design.

Unit 1:

GPU programming Model: introduction to CUDA, CUDA Execution Model.

Lab 0: Work through simple CUDA example, Synchronization, CUDA debugging and profiling tools.

Lab 1: Debug and profile a simple kernel before optimizing it.

Unit 2:

Lab 2: programming assignment of simple and tiled matrix multiplication in.

Test 1: Quiz based on descriptive questions.

Unit 3:

Lab 3: programming assignment of optimizing reduction tree. (memory specific optimizations)

Lab 4 (optional): Optimal binding of CPU cores and GPUs to a process.

Lab 5 (optional): Optimal mapping of CPU cores to GPUs in MPI-CUDA based applications.

Unit 4:

Course Project: This is a project intensive course, where in the student groups will apply the performance optimization techniques learned here to a chosen application.

Note: Depending on the circumstances, OpenCL could be used in place of CUDA. (OpenCL is another language similar to CUDA which is gaining importance recently).

TEXT BOOK

1. David B Kirk and Wen-mei W.Hwu, *Programming Massively Parallel processors, A hands on Approach*, Third Edition, Elsevier Morgan Kauffman, 2017.
2. David B Kirk and Wen-mei W.Hwu, *Programming Massively Parallel processors, A hands on Approach*, Second Edition, Elsevier Morgan Kauffman, 2012.
3. Annual nVIDIA GTC Conference Presentations on Performance Optimizations and toolset.

REFERENCES:

1. Jason Sanders and Edward Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison Wesley Professional, 1st edition, 2010.

2. Wen-mei W. Hwu, *GPU Computing Gems*, Emerald Edition, Morgan Kauffman Publishers, 2011.
3. Benedict R. Gaster, et al., *Heterogeneous Computing with OpenCL*, Morgan Kaufmann, 2012

* * *

MPC 2 CLOUD COMPUTING (3 Credits) (42 periods)

Course Objectives: This course deals with the theory of cloud computing, how cloud services can be built, and also techniques for security and privacy in such environments. The course provides a deep study of the importance of OS virtualization and how this core technique led to the evolution of large scale compute, storage and data processing in public cloud. The course will also introduce core physical infrastructural concepts such as the design of data centers for data and compute virtualization, data and compute collocation and disaggregation, and the concept of building reliability and resiliency at physical and virtual layers. The course also introduces the unique modifications required to achieve security and privacy in a multi-tenanted environment, and the role of compliance such as HIPAA, SOX, and GDPR. The notion of a private cloud that is deployed on-premise and the push towards hybrid and multi-cloud as a way to circumvent cloud reliability issues will also be presented. Completing this course will enable the student to be able to reason, compare and contrast various commercial cloud providers.

Course Outcomes: At the completion of the course the student will be able to Understand the fundamentals of cloud computing systems

- Understand the critical role played by operating system virtualization
- Understand various types of services offered in cloud systems
- Understand the unique nature of security and privacy in public cloud systems
- Understand the practical challenges in moving from on-premise to cloud
- Understand the fundamental techniques for storage, compute, networking and data processing in cloud
- Understand how cloud infrastructure and cloud data centers are designed
- Compare and contrast various commercial cloud systems

Course Syllabus:

Unit 1: What is Cloud Computing?	2 periods
Unit 2: Cloud Resource Management: Virtualization, Resource Allocation, Sharing and Scaling	6 periods
Unit 3: Cloud Infrastructure: Data Centers, Campus, Metros and Availability Zones	6 periods
Unit 4: Cloud Services: Iaas, PaaS, SaaS	6 Periods
Unit 5: Designing Cloud Services	4 periods
Unit 6: Private Cloud, Public Cloud, Hybrid Cloud, Multicloud	2 periods

Unit 7: Privacy, Security, Data Governance and Compliance

	4 Periods
Unit 8: Moving Applications, Services and Data to Cloud	2 Periods
Unit 9: Cloud Storage	2 Periods
Unit 10: Cloud Databases	2 Periods
Unit 11: Commercial Cloud Systems: AWS Case Study	2 Periods
Unit 12: Commercial Cloud Systems: Azure Case Study	2 periods
Unit 13: Commercial Cloud Systems: Google Cloud Case Study	2 Periods
Total:	42 Periods

SUGGESTED READINGS:

Unit 1 references:

1. **Above the Clouds** -https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

Unit 2 references:

1. OS Virtualization - https://www.cs.bgu.ac.il/~osce151/wiki.files/Virtualization.ppt
2. What is Virtualization? - https://www.ibm.com/cloud/learn/virtualization-a complete-guide

Unit 3 references:

1. Cloud Computing and Data Center Networking - http://www.cs.columbia.edu/~sambits/Lecture1-090810.ppt
2. Data Center Network Architectures - https://people.csail.mit.edu/alizadeh/courses/6.888/slides/lecture2.pdf
3. Data Center TCP - https://people.csail.mit.edu/alizadeh/papers/dctcpsigcomm10.pdf
4. TIMELY: RTT-based congestion control for Data Center https://people.csail.mit.edu/ghobadi/papers/timely_public_review_sigcomm_2015.pdf

Unit 4 references:

1. IaaS, PaaS, and SaaS - https://www.bmc.com/blogs/saas-vs-paas-vs-iaaswhats-the-difference-and-how-to-choose/

Unit 5 & 6 references:

1. https://www.oreilly.com/library/view/architectingthe-cloud/9781118826461/

Unit 7 references:

1. Guidelines on security and privacy on public cloud computing -

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf
2. Data Security and Privacy Protection in Public Cloud- https://arxiv.org/pdf/1812.05745.pdf

Unit 8 & 9 references:
1. Map Reduce: Simplified Data Processing on Large Clusters - https://research.google/pubs/pub62/
2. ADLS -
3. S3 -

Unit 10 references:
1. Dynamo DB - https://aws.amazon.com/dynamodb/
2. Spanner - https://static.googleusercontent.com/media/research.google.com/en//archive/spanner-osdi2012.pdf
3. RedShift - https://event.cwi.nl/lsde/papers/p1917-gupta.pdf

Unit 11, 12, 13: references:
1. AWS - https://aws.amazon.com/blogs/aws/whitepaper-on/
2. Azure Data Lake Storage- http://www.cs.ucf.edu/~kienhua/classes/COP5711/Papers/MSazure2017.pdf
3. BigTable key-value store- https://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf
4. Spanner Distributed Database - https://static.googleusercontent.com/media/research.google.com/en//archive/spannerosdi2012.pdf

MPC 2(P) Practicals: CLOUD COMPUTING (1 Credit) (28 periods)

Course Objective: To expose and develop practical understanding on the topics studied in the theory course such as Virtual Machines, Clusters and data Centers, Cloud Architecture and Service Models. This course aims to impart the practical skill pertaining AWS cloud computing environment, infrastructure and core services.

Course Outcome: At the end of the course a student will be able to
- Identify the global infrastructure components of AWS
- Create a virtual private cloud (VPC) by using Amazon Virtual Private Cloud (Amazon VPC)
- Demonstrate when to use Amazon Elastic Compute Cloud (Amazon EC2), AWS Lambda, and AWS Elastic Beanstalk
- Differentiate between Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), and Amazon Simple Storage Service Glacier (Amazon S3 Glacier)

- Demonstrate when to use AWS database services, including Amazon Relational Database Service (Amazon RDS), Amazon Dynamo DB, Amazon Redshift, and Amazon Aurora
- Explain the architectural principles of the AWS Cloud
- Describe the security and compliance measures of the AWS Cloud, including AWS Identity and Access Management (IAM)

Syllabus:

Introduction to Amazon Web Services(AWS) through hands on

- AWS Cloud Adoption Framework
- AWS Global Infrastructure
- AWS Regions, Availability zones, and edge locations
- AWS Services and broad service categories AWS Compute
- Demonstration of AWS EC2, Elastic BeanStalk, AWS Lambda, etc. AWS Storage : Introduction to different storage services which include:
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service Glacier Databases : The module describes the following database services:
    - Amazon Relational Database Service (Amazon RDS)
    - Amazon DynamoDB
    - Amazon Redshift
    - Amazon Aurora

Networking and Content Delivery: This module introduces three fundamental AWS etworking and content delivery services: Amazon Virtual Private Cloud (Amazon VPC), Amazon Route 53, and Amazon CloudFront. AWS Cloud Security: This module includes the controls in the AWS environment like

- Security Groups, IAM users and Groups
- AWS shared responsibility model
- AWS compliance programs
- Auto Scaling and Monitoring

Reference Text:

1. Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, Original English language edition copyright © 2012 by Elsevier Inc.

2. Amazon Cloud practitioner certification material available as academic training module on AWS site.

* * *

STREAM VII: SOFTWARE ENGINEERING

SE 1 OBJECT ORIENTED SYSTEM DESIGN

(3 Credits) (42 Periods)

Course Objectives:

Understanding the role played by the concept of "Objects" in computer science. Students are already aware of the ease in programming due to the concept of "object(s)". In this course a student also learns how even operating systems and software design can be understood better by this concept.

Course outcomes:

1) Learning object oriented design tools
2) Object oriented software design
3) System analysis based on object orientation

Course Syllabus:

INTRODUCTION

Overview of Object Oriented Systems Development - Object Basics: The object Model - Classes and Objects - Complexity - Notation - Process - Object types - Object state – Object - Oriented Systems Development Life Cycle. (4 Periods)

OBJECT ORIENTED METHODOLOGIES

Rumbaugh methodology - Booch methodology – Jacobson methodology –Patterns – Frameworks – Unified approach – Unified Modeling Language - Use case – Class diagram – Interactive diagram – Package diagram – Collaboration diagram – State diagram– Activity Diagram. (10 Periods)

OBJECT ORIENTED ANALYSIS:

Identifying use cases – Object analysis – Classification – Identifying object relationships – Attributes and methods. (8 Periods)

OBJECT ORIENTED DESIGN

Design axioms – Designing classes – Access layer– Object storage – Object interoperability. (12 Periods)

SOFTWARE QUALITY AND USABILITY

Designing interface objects–Software quality assurance– System usability–Metrics. (8 Periods)

Total 42 Periods

Reference Texts

1. Ali Bahrami, *Object Oriented Systems Development*, Irwin McGraw – Hill,1999.
2. Martin Fowler, *UML Distilled*, IInd Edn, PHI / Pearson Education, 2002.

Suggested Readings:

1. Grady Booch, *Object Oriented Analysis and Design with Applications*, IInd Edn, Benjamine Cummings, USA, 1994.
2. James R. Rumbaugh, Michael R. Blaha et al, *Object Oriented Modeling and Design*, Pearson Education Asia, 1991.
3. Betrand Meyor, *Object Oriented Software Construction*, IInd Edn, Prentice Hall PTR, New Jersey, 1997
4. Stephen R Schach, *Introduction to Object Oriented Analysis and Design*, Tata McGraw-Hill, 2003.
5. Tom Pender, *UML 2 Bible*, Wiley Publishing, Inc., 2005

SE 1(P) Practicals: OBJECT ORIENTED SYSTEM DESIGN (1 Credit) (28 periods)

Course Objective: Use state diagrams, activity diagrams to model the functioning of a system. Use UML diagrams to depict the various divisions involved in making a function happen.

Course Outcomes:

1) Learn UML diagrams
2) Learn Activity diagrams
3) Learn Use case analysis
4) Learn collaboration diagrams.

Syllabus:

Relevant exercises from different units in the syllabus will be implemented in the Lab.

Reference Texts

1. Ali Bahrami, *Object Oriented Systems Development*, Irwin McGraw – Hill,1999.
2. Martin Fowler, *UML Distilled*, 2nd Edition, PHI / Pearson Education, 2002.

* * *