File Ref.No.260197/GA - IV - J1/2021/Admn

UNIVERSITY OF CALICUT

Abstract

General and Academic IV - Faculty of Science - Syllabus of MSc Computer Science Programme for University teaching Department under CCSS PG Regulations 202- w.e.f 2022 admission -Implemented - Subject to ratification by the Academic Council -Orders Issued.

G & A - IV - J

U.O.No. 16292/2022/Admn Dated, Calicut University.P.O, 25.08.2022

Read:-1. U.O.No. 8481/2022/Admn dated, 11.04.2022
2. Item No. 1 in the minutes of the meeting of the Board of Studies in Computer Science & Application-PG dated 29.07.2022
3. Remarks of the Dean, Faculty of Science, dated 04.08.2022
4. Orders of the Vice Chancellor in the file of even no, dated 13.08.2022.

ORDER

1. The Regulations under Choice-based Credit Semester System for Post Graduate Programmes (CCSS PG-2022) of all Teaching Departments / Schools of the University of Calicut, has been implemented with effect from 2022 admission, vide paper read (1) above.
2. The meeting of Board of Studies in Computer Science & Application-PG held on 29.07.2022 has resolved to approve the revised scheme and syllabus of MSc Computer Science Programme in tune with Regulations under Choice- based Credit Semester System for Post Graduate Programmes (CCSS PG-2022) of all Teaching Departments / Schools of the University of Calicut, w.e.f. 2022 admission. vide paper read (2) above.
3. The Scheme and Syllabus of MSc Computer Science Programme (CCSS PG 2022), in tune with Regulations under Choice- based Credit Semester System for Post Graduate Programmes, has been approved by the Dean, Faculty of Science, vide paper read (3) above and by the Vice Chancellor, subject to ratification by the Academic Council, vide paper read (4) above.
4. The Scheme and Syllabus of MSc Computer Science Programme in tune with Regulations under Choice- based Credit Semester System for Post Graduate Programmes (CCSS PG-2022) of all Teaching Departments / Schools of the University of Calicut, w.e.f. 2022 admission, is therefore implemented, subject to ratification by the Academic Council.
5. Orders are issued accordingly. (Syllabus appended)

Abdussamad M

Assistant Registrar

To

1. Affiliated College
2. The Head, Department of Computer Science

Copy to: PS to VC/PA to PVC/ PA to Registrar/PA to CE/JCE I/JCE V/DoA/EX and EG Sections/GA I F/CHMK Library/Information Centres/IQAC/SF/DF/FC

Forwarded / By Order

Section Officer

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE

OBE Based Regulations, Course Structure, Scheme of Evaluation and Detailed Syllabus for

MASTER OF SCIENCE (M.Sc.)

in

COMPUTER SCIENCE

Choice Based Credit Semester System (CCSS)
(Effective from 2022 Admission Onward)

Under the

FACULTY OF SCIENCE

BOARD OF STUDIES IN COMPUTER SCIENCE AND APPLICATIONS (PG)
University of Calicut, Kerala 673 635

REGULATIONS

The existing regulations of Choice-based Credit Semester System (UO No. 4500/2019/Admn. Dated 26-03-2019) which are applicable for University Teaching Departments are also applicable for this programme with the following exceptions.

I **Objectives:** The course of the MSc (Computer Science) Programme is designed with the following objectives:

- To equip students to take up challenging research oriented responsibilities and courses for their higher studies/profession.
- To train and equip the students to meet the requirements of the Software Industry in the country and outside.
- To motivate and support the students to prepare and qualify challenging competitive examinations such as JRF/NET/JAM/GATE etc.

II **Programme Structure:**

1 Duration of the course shall be 2 years, divided into 4 semesters. The entire period of the four semester shall be divided for one core and one elective courses and for the Project Work.

2 The programme includes four types of courses, viz., Core courses (Code C), Elective Courses (Code E), Open Elective (Code O) and Audit Courses (Code A).

3 Selection and Eligibility for Admission is based on the existing University rules.

4 Evaluation of all semester theory papers will be on the basis of existing CCSS norms.

5 Conduct of Practical Examinations: Odd semester Practical Examinations will be conducted internally by the Department and Even Semester Examinations will be conducted by the Controller of Examination.

6 Core Courses: 'Core Course' means a course of a particular degree programme, which must be successfully completed by a student to receive the degree and which cannot be substituted by any other course. Core courses are offered by the Department conducting the programme.

7 Elective courses: An optional course to be selected by a student out of such courses offered in the same Department.

8 Open Elective Course: An elective course which is available for students of all programmes including students of the same department. Students of other departments will opt these courses subject to fulfilling of eligibility of criteria as laid down by the Department offering the course.

9 Audit Courses: There will be two Audit Courses with 2 credits each

i. Ability Enhancement Course: A compulsory course which is mandatory for all programmes but its grade will be not be counted for the computation of SGPA or CGPA. This course is to get professional competency and exposure in the coarsened core area.

ii. Professional Competency Course: A compulsory course which is mandatory for all programmes but its grade will be not be counted for the computation of SGPA or CGPA. This course is to enhance the ability and skill in the coarsened core area.

These have to be done one each in the first two semesters. The credits will not be counted for evaluating the overall SGPA and CGPA. Concerned Department shall conduct examination for these courses and have to intimate /upload the results of the same to the University on the stipulated date during the III Semester. Students have to obtain only minimum pass requirements in the Audit Courses.

10 **Project Work and Viva:** The Project work should be carried out over the period of 16 weeks in the final semester in an Industry / R and D organization / Department/Institution. If the project is carried out in an Industry / R and D organization outside the campus, then a co-guide shall be selected from the Department/ Institution concerned. Every student should do the Project individually and no grouping is allowed. All the candidates are required to get the approval of their synopsis and the guide before commencement of the project from the Department / Institution and the matter may be intimated to the University at the beginning of the semester by the Department / Institution. The project will be reviewed periodically every month by the Department / Institutional. The continuous assessment marks (CA) will be based on the periodic progress and progress report.

At the end of the semester the candidate shall submit the Project report (two bound copies and one soft copy) duly approved by the guide, co-guide for End Semester Assessment. Evaluation for ESA should be conducted by a board of examiners appointed by the University. (Mark Distribution: Content 30% + Methodology 30% + Presentation 20 %, and Via- voce 20 %). If project work and the report are found to be

not up to the expected standard, the examiners can ask the candidate to modify and resubmit the project report after incorporating the suggestions of the examiners. Such reports shall be resubmitted within the stipulated period suggested by the examiner(s).

A PROGRAMME OUTCOME (PO)

At the end of post graduate program at Department of Computer Science, University of calicut, a student would have:

PO1: Attained profound expertise in computing science discipline.

PO2: Acquired ability to function in multidisciplinary domains.

PO3: Equipped with comprehensive knowledge and understanding of advanced theoretical fundamentals in computer science as well as contemporary key research issues in specialized areas.

PO4: Attained ability to exercise research intelligence and innovations.

PO5: Proficiency in ethical principles and committed to professional ethics.

PO6: Incorporated self-directed and life-long learning.

PO7: Obtained ability to maneuver in diverse contexts with global perspective.

B PROGRAMME SPECIFIC OUTCOME (PSO)

PSO1: Evaluate complex real world problems by applying principles of theoretical computing, engineering and mathematical models.

PSO2: Understand concepts and applications in the field of Computer Sciences viz. Computational Intelligence, Machine learning, Web technology, Information Retrieval Systems, Data Analytics, Communication and networking.

PSO3: Identify, analyse, and synthesize scholarly literature relating to the field of computer science to design, analyze and interpret data to find solutions

PSO4: Conceive Project Management capabilities to solve real world problems in accordance with the needs of the industry, in a stipulated time frame.

PSO5: Prepare to address the challenging requirements coming from the enterprise applications.

PSO6: Innovate, experiment and analyze research findings and practice the process of scientific publishing.

UNIVERSITY OF CALICUT
Department of Computer Science
M.Sc. Computer Science
Under CCSS (Effective from 2022 Admission Onwards)
COURSE STRUCTURE AND SCHEME OF EVALUATION

Semester 1

Sl.No.	Course Code	Course	Instructional Hrs/week		Exam Duration		Marks			Credit
			Lect/Lab	Tutorial	Theory	Practical	ESA	CA	Total	
1	CSC1C01	Discrete Mathematical Structures	3	1	3	—	50	50	100	4
2	CSC1C02	Advanced Data Structures and Algorithms	3	1	3	—	50	50	100	4
3	CSC1C03	Principles of Programming Methodology	3	1	3	—	50	50	100	4
4	CSC1C04	Theory of Computation	3	1	3	—	50	50	100	4
5	CSC1C05	Computer Organization and Architecture	3	1	3	—	50	50	100	3
6	CSC1C06	Practical 1	10	5	—	3	50	50	100	3
7	CSC1A01	Introduction to Research Methodology	—	—	3		—	100	100	2
	Total		25	10	—	—	—	—	600	22

Semester 2

Sl.No.	Course Code	Course	Instructional Hrs/week		Exam Duration		Marks			Credit
			Lect/Lab	Tutorial	Theory	Practical	ESA	CA	Total	
1	CSC2C07	Design and Analysis of Algorithms	3	1	3	—	50	50	100	4
2	CSC2C08	Advanced Database Management System	3	1	3	—	50	50	100	4
3	CSC2C09	Operating System and Virtualization	3	1	3	—	50	50	100	3
4	CSC2C10	Object Oriented Programming with Java	3	1	3	—	50	50	100	4
5	CSC2E01-05	Elective 1	3	1	3	—	50	50	100	4
6	CSC2C11	Practical 2	10	5	—	3	50	50	100	3
7	CSC2A02	Term Paper	—	—	—	—	—	100	100	2
	Total		25	10	—	—	—	—	600	22

Elective 1	Credit
CSC2E01- Artificial Intelligence	4
CSC2E02 - Pattern Recognition	4
CSC2E03 - Computer Graphics	4
CSC2E04 - Cryptography and Network Security	4
CSC2E05 - Simulation and Modeling	4

Semester 3

Sl.No.	Course Code	Course	Instructional Hrs/week		Exam Duration		Marks			Credit
			Lect/Lab	Tutorial	Theory	Practical	ESA	CA	Total	
1	CSC3C12	Principles of Compiler Design	3	1	3	—	50	50	100	4
2	CSC3C13	Data Communication and Networking	3	1	3	—	50	50	100	3
3	CSC3C14	Web Technology	3	1	3	—	50	50	100	4
4	CSC3E06-CSC3E10	Elective 2	3	1	3	—	50	50	100	4
5	CSC3E11-CS3E14/CSC3O 01	Elective 3	3	1	3	—	50	50	100	4
6	CSC3C15	Practical 3	10	5	—	3	50	50	100	3
	Total		25	10	—	—	—	—	600	22

Elective 2	Credit
CSC3E06 - Digital Image Processing	4
CSC3E07 - Digital Speech Processing	4
CSC3E08 - Natural Language Processing	4
CSC3E09 - Bioinformatics	4
CSC3E10 - Computer Vision	4

Elective 3	Credit
CSC3E11 - Data Analytics Using Python	4
CSC3E12 - Data Mining	4
CSC3E13 - Information Retrieval System	4
CSC3E14 - Distributed Systems and Parallel Computing	4
Open Elective	Credit
CSC3O 01 - Introduction to Data Science	4

Semester 4

Sl.No.	Course Code	Course	Instructional Hrs/week		Exam Duration		Marks			Credit
			Lect/Lab	Tutorial	Theory	Practical	ESA	CA	Total	
1	CSC4C16	Software Engineering	3	1	3	—	50	50	100	4
2	CSC4E15-21	Elective 4	3	1	3	—	50	50	100	4
			Duration of the Project				Viva voce	CA	Total	

3	CSC4C17	Project Work & dissertation	16 Weeks	—	—		50	50	100	14
	Total				—	—	—	—	300	22

Elective 4	Credit
CSC4E15 - Advanced Machine Learning	4
CSC4E16 - Big data Technologies	4
CSC4E17 - Mobile Communication	4
CSC4E18 - Internet of Things	4
CSC4E19 - Cyber Physical Systems	4
CSC4E20 - Block chain Technology	4
CSC4E21 - Quantum Computation	4

UNIVERSITY OF CALICUT

Department of Computer Science

M.Sc. Computer Science
Under CCSS (Effective from 2022 Admission Onwards)

DETAILED SYLLABUS

First Semester

CSC1C01: Discrete Mathematical Structures

Course Number: 1
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Verify the validity of an argument using propositional and predicate logic

CO2: Understand the basic concepts of set theory and apply operations on set

CO3: Apply operations of relations and functions in discrete structures

CO4: Identify Group, Ring and Field in Group Theory

CO5: Understand applications of Graph Theory and Tree

CO6: Employ the concepts of graph theory and trees to formulate problem solving

Unit I :

Prepositional Logic: Statement Formulas and Truth Tables, Well Formed Formulas-Tautologies-Equivalence of Formulas-Duality Law-Tautological Implications-Normal Forms, Theory of Inference for the Statement Calculus. Predicate Calculus, Quantifiers, Free and Bound Variables, Inference Theory of the Predicate Calculus.

Unit II :

Set Theory-Sets and subsets-Set operations and their properties-Cartesian Products, Relations-Relation matrices- Properties of relations-Composition of relations, Equivalence relations and partitions-Partial Ordering.

Unit III :

Functions and Relations: Functions-Types of Functions, Composition of Functions and Inverse Functions. Relations -Relations and Their Properties, Functions as relations, Closure of Relations, Composition of relations, Equivalence Relations and Partitions. Partial Ordering, Hasse Diagram. The Pigeon hole Principle.

Unit IV :

Group Theory-Definition and Elementary Properties-Cyclic Groups- Homomorphism and Isomorphism-Subgroups-Cosets and Lagrange's Theorem, Rings and Fields-Definitions and examples of Rings, Integral Domains and Fields

Unit V :

Graph Theory-Paths and Cycles, Graph Isomorphism, Bipartite Graphs, Subgraphs, Representation of Graphs, Eulerian and Hamiltonian Properties of Paths-Trees-Spanning Trees, Cayley's theorem, Kruskal's Algorithm, Prim's Algorithm. Dijkstra's Algorithm to Find Shortest Path in Weighted Graphs.

References:

1. Trembley J.P. & Manohar R.P, Discrete Mathematical Structures with Application to Computer Science, Mc.Graw Hill, 2007
2. R.P.Grimaldi, Discrete and Combinatorial Mathematics: An applied Introduction, 3/e, Addision-Wesley, New Delhi, 1994
3. J.K.Truss, Discrete Mathematics for Computer Scientists, Addison Wesley, 1999
4. B.Kolman and R.C.Busby, Discrete Mathematical Structures for Computer Science, PHI, 1994
5. C.L.Liu. Elements of Discrete Mathematics, 2/e, McGraw Hill, 1985
6. John O. Clark , Derek A. Holton, A First Look at Graph Theory, world scientific publishing, 1995.

CSC1C02: Advanced Data Structures and Algorithms

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand different categories of data structures

CO2: Design algorithms to perform operations with linear and non – linear data structures.

CO3: Describe how arrays, linked lists, stacks, queues, trees and graphs are represented in memory and used by algorithms.

CO4: Describes various types of trees and heap structures and different methods for traversing trees.

CO5: Understand the hashing algorithms and their implementations.

CO6: Discuss the computational efficiency of the principal algorithms for sorting, searching and hashing.

Unit I :

Overview of Data Structures, Data Abstraction & Abstract data types. Arrays-Records-Representation. Data Structure operations: Traversing, Inserting and deleting, sorting and searching. Linear Search & Binary Search- Complexity.

Unit II :

Linear Data structures: Stack-operations and its implementations-Parsing arithmetic expressions, translating and evaluating; Recursion- characteristics of recursion-comparison of recursive and non-recursive algorithms, Queue- operations and its implementations-Circular queue-dequeue - priority queues, Linked Lists- Operations and implementations-Doubly Linked Lists and Circular lists-Sparse matrix representation.

Unit III :

Non-linear Data Structures: Trees-Tree traversals algorithms- Binary Tress-Threaded Binary Trees-Binary search Trees-Traversals and operations on BST - balanced trees-AVL, Red-Black Trees, Splay Tree, B-Tree, M-way Trees-Operations and their implementation.

Unit IV :

Hashing: Overview of hashing-Hash tables- hash functions and their computations-open addressing-Linear probing- quadratic probing-double hashing algorithms and their implementations-Separate chaining-Hashing efficiency.

Unit V :

Heap: Overview of heaps-Implementation and operations. Sorting techniques: Insertion sort- Selection sort- Shell sort- Bubble sort- Quick sort-Heap sort- Merge sort- External sort- Comparison of sorting algorithms. Graphs- representation of graphs-operations- traversals and their implementation- minimum spanning trees-shortest path problem-Efficiency of various graph algorithms.

References:

1. Alfred V.Aho, John E.Hopcroft and Jeffrey D.Ullman, Data structures and Algorithms,Pearson Education Asia,2002.
2. Horowitz E & Sahni S, Fundamentals of data structures, Computer Science press, 1978.
3. SartajSahni, Datastructures, Algorithms and Applications in Java, Second Edition, Universities Press(India) Pvt Ltd, 2005.
4. Robert Lafore, Datastructures and algorithms in Java, Second Edition, Sams Publishing,2003.
5. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein, Introduction toAlgorithms,Third Edition, PHI,2010.
6. Seymour Lipschutz and GAV Pai, Data Structures, Indian Adapted Edition, Schaum's Outlines Series, TMH, 2006

CSC1C03: Principles of Programming Methodology

Course Number: 3
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Illustrate various notions of flowchart and algorithms.

CO2: Determine the data representation formats for a specific problem domain.

CO3: Demonstrate C Language preliminaries, Operators and expressions

CO4: Examine the merits and demerits of various programming constructs viz pointers structures and union

CO5: Implement the basic operations in file handling

CO6: Design, develop, implement, test and document well-structured and reliable computer programs using the C programming language.

Unit I :

Problem Solving-Flow Chart for Structured Programming-Program Charts-System Charts-Variables, data names, programming statements-Flow Chart Symbols-Terminal Symbols-I/O-Comments- Connectors-Process-Decision. Algorithm Design-Problem Solving Aspect- Top Down Design-Formal Conventions- Writing Algorithms- Fundamental Algorithms with flow chart. Program- Characteristics of a good program- Modular Approach-Programming style- Documentation and Program Maintenance- Compilers and Interpreters- Running and Debugging Programs- Syntax Errors-Run- Time Errors- Logical Errors- Concept of Structured Programming.

Unit II :

C Language preliminaries-C character set, Identifiers and keywords, Data types, Declarations, Expressions, statements and symbolic constants. Input-Output-getchar, putchar, scanf, printf, gets, puts, functions. Preprocessor directives. Operators and expressions-Arithmetic, unary, logical, bit-wise, assignment and conditional operators. Control statements-While, do-while, for statements, nested loops, if else, switch, break, Continue, and goto statements, comma operators. Storage types-Automatic, external, register and static variables.

Unit III :

Functions-Defining and accessing, passing arguments, Function prototypes, Recursion, Library functions, Static functions. Arrays-Defining and processing, Passing arrays to a function, Multi-dimensional arrays. Strings-Defining and operations on strings.

Unit IV :

Pointers-Declarations, Passing pointers to a function, Operations on pointers, Pointer Arithmetic, Pointers and arrays, Arrays of pointers function pointers. Structures-Defining and processing, Passing to a function, Unions, typedef, array of structure, and pointer to structure.

Unit V :

File structures-Definitions, concept of record, file operations: Storing, creating, retrieving, updating Sequential, relative, indexed and random access mode, Files with binary mode(Low level), performance of Sequential Files, Direct mapping techniques: Absolute, relative and indexed sequential files (ISAM) concept of index, levels of index, overflow handling. File Handling-File operation-creation, copy, delete, update- Text file, binary file.

References:

1. Martin M. Lipschutz and Seymour Lipschutz, Schaum's Outline of Theory and Problems of Data Processing.
2. Anil Bikas Chaudhuri, The Art Of Programming Through Flowcharts & Algorithms, Laxmi Publications, New Delhi.
3. Jean Paul Trembley and Pual G Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill.

CSC1C04: Theory of Computation

Course Number: 4
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand basic concepts in the theoretical foundations of computer science.

CO2: Interpret types of formal languages and its machine equivalence

CO3: Construct automation and grammar for all formal languages

CO4: Model the concept of Turing Machines and Develop mathematical views towards general computation.

CO5: Understand machines hierarchy with respect to the capabilities using the Chomsky hierarchy.

CO6: Classify NP and P problems

Unit I :

Preliminaries: Review of proof techniques-Mathematical induction - Basic concepts of languages automata and grammar-Alphabet, languages and grammars, productions and derivation. Regular languages: Regular expressions- Finite deterministic and non-deterministic automata-regular grammar. Equivalence between various models. Closure properties regular languages-DFA state minimization-Pumping lemma and proof for existence of non regular languages.

Unit II :

Context-Free Grammars (CFG)- Derivations- sentential forms- Parse tree- Ambiguity in grammars and Languages- Applications of CFG- Simplification of Context free Grammars- Normal forms: Chomsky Normal form (CNF) and Greibach Normal form (GNF).

Unit III :

Pushdown Automata (PDA)- Formal definition-Graphical notations - Language accepted by PDA- Deterministic and Non Deterministic PDA - Equivalence of PDAs and CFGs-Pumping lemma for CFLs, Closure properties of CFLs - Decision properties of CFL. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

Unit IV :

Turing Machines- Notation- Instantaneous Description- Transition Diagram- The language of a Turing Machine- Variants of TMs-Multitape TMs, Nondeterministic TMs- TMs with semi - infinite tapes, multistack machines - Universal Turing Machines- Equivalence of the various variants with the basic model - Church-Turing Thesis.

Unit V :

Computability- Closure properties of recursive and recursively enumerable language. Context Sensitive Language and LBA- Equivalence of LBA and CSG-The Chomsky Hierarchy Un-decidability - Halting problem-reductions-Complexity: Complexity Classes - Class P - Class NP- NP complete and NP Hard problems.

References:

1. Linz: P. An Introduction to Formal Languages and Automata, Narosa, 1998.
2. HoporoftJ.E.and Ullman J.D., Introduction to Automata Theory Languages and Computation, Narosa, 1998.
3. H.R.Lewis and C.H.Papadimitriou, Elements of the Theory of Computation, Prentice Hall of India, 1996.
4. Martin J.C., Introduction to Languages and the Theory of Computation, Tata McTraw Hill, 1997.
5. J.E.Sagage, Models of Computation, exploring the power of Computing, Addison Wesley, 1998.
6. Michael Sipser : Introduction to theory of Computation , Cenage Learning, Indian Edition
7. D. S. Garey and G. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

CSC1C05: Computer Organization and Architecture

Course Number: 5
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 3
Course Outcomes:

CO1: Model number system concepts.

CO2: Design sequential circuit and combinational circuit.

CO3: Demonstrate the functional architecture of a computer system.

CO4: Illustrate principles of the memory system in detail.

CO5: Understand basic Input / Output Organization,interrupt processing and I/O interfaces.

CO6: Understand 8085 and 8086 microprocessor architecture.

Unit I :

Digital Computers and Digital Systems, Number systems and Conversions, Digital Logic Circuits- Boolean Algebra & Map simplification, combinational circuits, flip flops, design of sequential circuits, Digital Components-Integrated Circuits, Decoders, Multiplexers, Registers, Shift Registers, Binary Counters, data representation - data types, complements, fixed-point representation, floating-point representation, other binary codes, error detection codes.

Unit II :

Computer Arithmetic: addition of positive numbers, design of fast adders, signed addition and subtraction, Arithmetic and Branching Conditions, Multiplication of positive numbers, signed number multiplication- Booth Algorithm, fast multiplication, division- restoring and non-restoring algorithms, floating point numbers and operations- single & double precision.

Unit III :

Basic computer organization-machine instructions- classification, function, addresses, size, addressing modes- instruction cycle - instruction sequencing. fundamental concepts- registers, register transfers, performing arithmetic or logic operations, memory read and write, execution of a

complete instruction, branch instruction, Single bus, two bus, three bus organization, a complete processor- Control unit: - hardwired control, microprogrammed control, micro instructions-types.

Unit IV :

The Memory System: Some Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Virtual Memories, Memory Management Requirements, Secondary Storage, memory interleaving. Input / Output Organization -Accessing I/O devices- programmed I/O, interrupt I/O - interrupts - interrupt processing- hardware interrupts-programmable interrupt controller-vectored interrupts - interrupt nesting - daisy chaining- direct memory access (DMA)- DMA operations & DMA Controller, Introduction to I/O interfaces, I/O channels, IO Processors.

Unit V :

8085 microprocessor - Architecture: Block diagram-addressing modes-instruction set, Instruction cycle-timing diagrams - different machine cycles - fetch and execute operations - estimation of execution time. 8086 microprocessor - Architecture: Block diagram-Intel 8051 Micro controller-Architecture - basic instructions-basic assembly language programs-peripherals: interrupts, timers, parallel port, serial port.

References:

1. V C Hamacher, Computer Organization, Mc-Graw Hill International Edition, Fifth Edition.
2. Morris Mano, Digital logic and Computer design, Prentice Hall of India, 2004.
3. M Morris Mano, Computer System Architecture, Prentice Hall, Third Edition.
4. William Stallings, Computer Organization and Architecture, Fifth Edition.
5. Andrew S Tanenbaum, Structured Computer Education, Prentice Hall, Fourth Edition.
6. Floyd and Jain, Digital Fundamentals, Pearson Education, Eighth Edition.
7. Ramesh. S. Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Wiley Eastern Ltd, New Delhi.

CSC1C06: Lab I - C Programming

Course Number: 6
Number of Credits: 3
Number of Contact Hours: 5 (10 Practical + 5 Tutorial)
Course Outcomes:
Develop programs relating to the theory portions covered in CS1CC03 Principles of Programming Methodology and CS1CC02 Advanced Data Structures and Algorithms

CO1: Develop programming skills using the fundamentals and basics of C language.

CO2: Develop programs using the basic elements like control statements, arrays and strings.

CO3: Design and implement the effective usage of arrays, structures, functions and pointers.

CO4: Implement files handling and command line arguments.

CO5: Demonstrate the concepts of stack, queue and linked list and apply various operations on them.

CO6: Demonstrate the concept of sorting and searching techniques, tree traversal and its operations.

Unit I C Programming:

Develop programs to implement the following:

1. HCF (Euclid?s algorithm) and LCM of given numbers.
2. Find mean, median and mode of a given set of numbers
3. Conversion of numbers from binary to decimal, hexadecimal, octal and back.
4. Evaluation of functions like ex, sinx, cosx etc. for a given numerical precision using Taylor?s series.
5. Testing whether a given number is prime.
6. String manipulation programs: sub-string search, deletion.
7. Lexicographic sorting of a given set of strings.
8. Generation of all permutations of the letters of a given string using recursion.

9. Programs to find the product of two matrices.
10. Inverse and determinant (using recursion) of a given matrix.

Unit II Data Structures and Algorithms:

Implement the following:

1. Singly linked list with operations to access data, add node and delete node.
2. Variations on linked lists.
3. Sparse matrix.
4. PUSH, POP operations of stack using Arrays/using linked lists.
5. Add, delete operations of a queue using Arrays.
6. Add, delete operations of a queue using linked lists.
7. Variations on queues.
8. Conversion of infix to postfix using stack operations.
9. Postfix Expression Evaluation using stack.
10. Towers of Hanoi Problem
11. Addition of two polynomials using linked list.
12. Binary tree using linked lists
13. Binary tree traversals.
14. Variations on tree structures
15. Graphs/Graph traversals
16. Shortest path algorithm
17. Sorting techniques/Search algorithms.

CSC1A01: Introduction to Research Methodology

Course Number: 7
Number of Credits: 2
Course Outcomes:

CO1: Understand foundations of research, characteristics of scientific methods.

CO2: Describe problem identification and formulation process.

CO3: Apply the ethical principles of research.

CO4: Identify the components of a literature review process, research design,data analysis and interpretation.

CO5: Familiarize the tools and techniques for Research and Critically analyze published research works.

CO6: Innovate and apply research methods in the discipline of computing.

Unit I :

Foundations of Research: Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method - Understanding the language of research - Concept, Construct, Definition, Variable. Research Process.

Unit II :

Problem Identification & Formulation - Research Question - Investigation Question - Measurement Issues - Hypothesis - Qualities of a good Hypothesis - Null Hypothesis & Alternative Hypothesis. Hypothesis Testing - Logic & Importance.

Unit III :

Research Design: Concept and Importance in Research - Features of a good research design - Exploratory Research Design - concept, types and uses, Descriptive Research Designs - concept, types and uses. Experimental Design: Concept of Independent & Dependent variables.

Unit IV :

Data Analysis: Data Preparation - Univariate analysis (frequency tables, bar charts, pie charts, percentages). Interpretation of Data and Paper

Writing - Layout of a Research Paper, Journals in Computer Science, Impact factor of Journals, When and where to publish ? Ethical issues related to publishing, Plagiarism and Self-Plagiarism.

Unit V :

Use of Encyclopedias, Research Guides, Handbook etc., Academic Databases for Computer Science Discipline. Use of tools / techniques for Research: methods to search required information effectively, Reference Management Software like Zotero/Mendeley, Software for paper formatting like LaTeX/MS Office, Software for detection of Plagiarism.

References:

1. Kothari, Chakravanti Rajagopalachari. Research methodology: Methods and techniques. New Age International, 2004.
2. Blumberg, Boris, Donald R. Cooper, and Pamela S. Schindler. Business research methods. Vol. 2. London: McGraw-Hill Higher Education, 2008.
3. Misra R.P, Research Methodology - A Hand Book, Concept publishing Company, New Delhi, 1988
4. Bell, Emma, Alan Bryman, and Bill Harley. Business research methods. Oxford university press, 2018.
5. https://nptel.ac.in/courses/121106007/

Second Semester

CSC2C07: Design and Analysis of Algorithms

Course Number: 1
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Design algorithm in context of space and time complexity and apply asymptotic notation

CO2: Model the correctness of algorithms using inductive proofs and invariants

CO3: Demonstrate the divide-and-conquer and Transform-and-Conquer techniques to explain when an algorithmic design situation calls for it.

CO4: Analyse the complexity of Greedy approach and Dynamic Programming

CO5: Interpret classes P, NP and NP Complete problems and familiarise problems in each category.

CO6: Demonstrate the use of parallel algorithms.

Unit I :

Algorithm Analysis: Steps in developing algorithms, Methods of specifying an algorithm. Important Problem Types. Importance of algorithm analysis, Time and Space Complexity. Growth of Functions: Asymptotic notations. Solving recurrences: Substitution method, Recursion Tree method, Master's Theorem, problem solving using Master's theorem.

Unit II :

Basic Designing Techniques: Brute Force Approaches: Brute Force String Matching. Divide-and-Conquer: Merge sort, Quick sort, Binary Search, Stressen's Matrix Multiplication. Greedy Method: The General Method, Huffman coding, Knapsack Problem, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm; Single Source Shortest Path algorithm.

Unit III :

Dynamic Programming: The General Method, Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem, 0/1 Knapsack, The Traveling Salesperson problem. Backtracking: n-Queens problem, Hamiltonian Circuit Problem, Sum of subset Problem. BranchandBound: Assignment Problem, Knapsack Problem, Traveling Salesperson Problem.

Unit IV :

Complexity: Complexity classes – P, NP, NP-Hard and NP-complete Problems , NP-completeness reductions for Hamiltonian cycle and Traveling Salesman Problem . P versus NP problem.

Unit V :

Design and Analysis of Parallel Algorithms: PRAM models – EREW, ERCW, CREW and CRCW, Relation between various models, Analyzing Parallel Algorithms: Time Complexity, Cost, Number of Processors, Space Complexity, Speed up, Efficiency, Scalability, Amdahl's Law. Parallel merging and sorting.

References:

1. Anany Levitin: Introduction to the Design & Analysis of Algorithms, 2nd Edition, Pearson Education, 2007.
2. Thomas H Cormen, Charles E Leiserson, & Ronald L Rivest, Introduction to Algorithms, 2nd Ed. Prentice Hall of India Private Limited, New Delhi, 2001.
3. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: Fundamentals of Computer Algorithms, 2nd Edition, Universities Press, 2007.
4. S. Basse, Computer Algorithms: Introduction to Design and Analysis, Addison Wesley, 1998.

5. U.Manber, Introduction to Algorithms: A creative approach, Addison Wesley, 1989
6. V.Aho, J.E. Hoperaft, J.D.Ullman, The design and Analysis or Computer Algorithms, Addison Wesley, 1974
7. Gilles Brassard & Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India, 2007.
8. Goodman S E & Hedetnieni, Introduction to the Design & analysis of Algorithms, Mc- Graw Hill, 2002.

CSC2C08: Advanced Database Management System

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Illustrate the basics of database system concepts, concepts of relational data model, entity-relationship model,

CO2: Understand the concepts of relational database design, relational algebra and calculus

CO3: Apply the normalization techniques to improve the database design.

CO4: Describe various database manipulation commands in SQL.

CO5: Understand Transaction Processing Locking using the concept of Concurrency control.

CO6: Conceptualize advanced features of Object-Oriented Database Management Systems and Distributed databases.

Unit I :

Database System Concepts, Purpose of Database Systems, Views of Data - Data Abstraction, Instances and Schemes, Data Independence, Data Models. Relational Data Model - Relational Model concepts, keys, Integrity constraints - Domain Constraints, Key Constraints, Entity Integrity Constraints , Referential Integrity Constraints. ER Data Model-Design Issues, Entity Relationship Diagram, Weak Entity Sets, Extended ER Features, Design of an ER Database Schema, Reduction of an ER Schema to Tables. Relational Algebra-Selection and Projection, Set operations, Renaming, Joins, Division. Relational Calculus.

Unit II :

Relational database design - Design Principles, Normalization, Normal Forms - 1NF, 2NF, 3NF, BCNF, 4NF & PJNF, Domain Key Normal Form. Transactions -concepts, states of Transactions, ACID properties. Schedules - serial schedules, concurrent schedules, Serializability, Concurrency control protocols - Locking-Two-Phase Locking, Deadlock, Granularity, Timestamp Ordering Protocol.

Unit III :

Relational Database Query Languages - Basics of QBE , SQL –Data Definition Language(DDL), Data Manipulation Language(DML), Data Control Language(DCL) Commands, Basics of Query Processing. Data Definition in SQL - Data types, Creation, Insertion, Viewing, Updation, Deletion of tables, Renaming, Dropping of tables. Data Constraints - I/O constraints, Primary key, foreign key, unique key constraints, Database Manipulation in SQL - Select command, Logical operators, Range searching, Pattern matching, Grouping data from tables in SQL, GROUP BY, HAVING clauses, Joins. Views - Creation, Renaming the column of a view, DROP view.

Unit IV :

Program with SQL - Data types: Using set and select commands, Flow control constructs - if, if /else, while, goto. Global variables, Data types, Operators and Functions, Data Definition and Manipulation Statements, CodeBlocks, Stored Procedures- create, alter and drop, passing and returning data to stored procedures, using stored procedures within queries. Cursors - Working with cursors, Error Handling. User defined functions, implementing triggers.

Unit V :

Distributed Database systems- characteristics, architecture and challenges. Different approaches in database technologies - Object oriented and Object relational databases. Emerging trends in databases. Introduction to Big Data.

References:

1. AbrahamSilbersehatz, Henry F. Korth and S.Sudarshan, Database system concepts,6th Edition, Tata McGraw-Hill 2010.
2. Elmasri and Navathe, Fundementals of Database systems, 5th Edition, Pearson, 2009.
3. CJ Date, Introduction to Database Systems, Addison Wesley.
4. VikramVaswani, MySQL The complete Reference,1st Edition, Tata McGraw-Hill, 2004.
5. Paul DuBois, MySQL Cookbook, 2nd Edition, O'Reilly Media, 2006

CSC2C09: Operating System and Virtualization

Course Number: 3
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 3
Course Outcomes:

CO1: Understand the basic components of a computer operating system.

CO2: Apply the concept of Process & thread management.

CO3: Understand the CPU Scheduling algorithms and the concept of Concurrency control.

CO4: Analyse the performance of memory allocation and replacement techniques.

CO5: Understand the concepts of protection and security in an OS

CO6: Understand the basics of Virtualization.

Unit I :

System software Overview: Operating system, I/O Manager, Assembler, Macro Processor Compiler and Interpreters, Linker, Loader. Fundamentals of OS: OS services and components, Different types of operating systems, multitasking, multiprogramming, time sharing, buffering, spooling.

Unit II :

Process & thread management: Concept of process and threads, process states, process management, context switching, interaction between processes and OS, multithreading. CPU Scheduling algorithms, Concurrency control: Concurrency and race conditions, mutual exclusion requirements , s/w and h/w solutions, semaphores, monitors, classical IPC problem and solutions , Dead locks -characterization , detection ,recovery, avoidance and prevention.

Unit III :

Memory management: Issues-Memory Allocation, Dynamic Relocation, various management strategies. Virtual memory, Paging: Issues and Algorithms. Segmentation: Typical implementations of paging and segmentation systems, Disk Scheduling. File Systems: File concept, File

support, Access methods, Allocation methods, Directory systems, File Protection, Free Space management.

Unit IV :

Protection & security-Protection: Goals of protection, Domain of protection, Access matrix, Implementation of access matrix, Revocation of access rights .Security: the security problem, authentication, one-time passwords, program threats, System threats, Threat monitoring, Encryption. Case study of Microsoft Windows XP.

Unit V :

Virtualization Concepts: Virtual machines; supporting multiple operating systems simultaneously on a single hardware platform; running one operating system on top of another. True or pure virtualization. Approaches to Virtualization: Processor Issue, Memory Management, I/O Management, Java VM, Linux VServer Virtual Machine Architecture, Android Virtual Machine.

References:

1. D. M. Dhamdhare, Operating Systems, Tata Mc Graw Hill, 2nd Ed.
2. Silberschatz & Galvin, Operating System Concepts, Wiley, 7th Ed.,2000.
3. Gary J Nutt, Operating systems-A Modern Perspective, Addison Wesley,2000.
4. Flynn & Metioes, Understanding Operating System, Thomsan, 4th Ed.
5. Andrew Tanenbam, Albert S. Woodhull, Operating Systems Design & Implementation, Pearson.
6. Operating Systems: Internals Design Principles, 8 th Edition, William Stallings, Pearson Education India
7. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
8. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)

CSC2C10: Object Oriented Programming with Java

Course Number: 4
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Model object oriented programming concepts and develop Java programs that include basic constructs

CO2: Design and implement object-oriented programs including packages and interfaces.

CO3: Understand exception handling and debugging.

CO4: Develop GUI and applets for web based applications.

CO5: Apply User Interface Components with Swing.

CO6: Describe socket programming, JDBC architecture and connectivity.

Unit I :

Introduction to Object Oriented Programming, Comparison with other programming paradigms, Java Basics: Java Programming environment, Structure of a java program, Life cycle of a java program, Java Virtual Machine, Byte code, Features of Java, fundamental programming structures in Java: comments, Primitive Data Types ,variables, Keywords, literals, variables scope & declarations, Control structures, Operators - Casts and Conversions, Arrays. Simple programs.

Unit II :

Data abstraction and Encapsulation - Objects and Classes: Predefined Classes - Defining Classes- Static Fields and Methods - Method Parameters - Object Construction- Packages. Inheritance: Classes - Super classes - Subclasses - Object: The Cosmic Superclass - Generic Array Lists -Object Wrappers and Auto-boxing - Reflection - Enumeration Classes. Interfaces and Inner Classes, Polymorphism: Overloading - Overriding.

Unit III :

Introduction to GUI: AWT Architecture - Light-Weight vs Heavy-Weight, AWT Event Hierarchy & Event Handling - Using Top-Levels - components and containers - Introduction to Layouts. Deploying Applets and Applications: Applet Basics - The Applet HTML Tags and Attributes - Multimedia - The Applet Context - JAR Files - Application Packaging.

Unit IV :

Exceptions and Debugging: Dealing with Errors - Catching Exceptions - Using Exceptions, user defined Exceptions. Threads - Creating Threads, Runnable interface, Thread Class, Inter thread communication, Synchronization suspending, resuming and stopping threads. Multithreaded Programming. Streams and Files: The Complete Stream - ZIP File Streams - Use of Streams - Object Streams - File Management.

Unit V :

User Interface Components with Swing: Introduction to Layout Management - Text Input Choice Components - Menus - Sophisticated Layout Management - Dialog Boxes. Java library. Database Programming: JDBC - introduction, architecture, Drivers, connections, statements, resultset, Meta data and Query Execution. Sockets: Introduction to networking, InetAddress, url, socket, server sockets, Datagrams.

References:

1. Horstmann & Coronell, "Core Java ", Volume 1 and 2, 8th Ed., Pearson, 2008.
2. Herbert Schildt, "Java2 The Complete Reference", Seventh Edition, Tata McGraw-Hill,2007.
3. Bruce Eckel, "Thinking in Java", Prentice Hall, Fourth Edition, 2006.
4. Kathy Sierra and Bert Bates, "Head first java", Second Edition, Oreilly Media, 2005.
5. Y.Daniel Liang, "Introduction to Java Programming", Eighth Edition, Pearson, 2011.
6. James. P. Cohoon, Jack. W. Davison, "Programming in java 5.0", Tata McGraw Hill, 2006.
7. Thomas Wu, "An introduction to Object Oriented Programming with Java", Tata McGraw Hill, 2006.
8. Bernard Van Haecke,"JDBC: Java Database Connectivity", IDG Books India, 2000.

CSC2C11: Practical 2

Course Number: 5
Number of Contact Hours: 5 (10 Practical + 5 Tutorial)
Number of Credits: 3
Course Outcomes: *Construct programs relating to the theory portions covered in CS1CC10 Object Oriented Programming with Java and CS1CC08 Advanced Database Management System*

CO1: Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs.

CO2: Design and develop Java programs that solve real-world problems and utilize Java Graphical User Interface in program writing..

CO3 Design and development of relational database systems.

CO4 Understand the use of various advanced queries execution such as relational constraints, joins, set operations, aggregate functions, trigger and views.

CO5 Describe the Use of various software to design and build ER Diagrams, UML, Flowchart for related database systems.

CO6 Apply JDBC to provide a program level interface for communicating with database using Java programming.

Unit I Java Programming:

Develop programs to implement the following:

1. Classes, objects and methods.
2. Inheritance of different types.
3. Use of keywords super, abstract and final.
4. Method overloading and Method overriding
5. Packages and interfaces.
6. Exception handling
7. Use of static members in a class.
8. File operations.
9. Multithreaded Programming
10. Applets

11. String handling
12. AWT to work with text and graphics
13. Applications of Swing.

Unit II Advanced DBMS:

Implement the following:

1. DCL, DDL, DML, DQL statements in MySQL
2. Stored Procedures in MySQL
3. Cursors and Triggers
4. Transactional and Locking statements
5. Account Management statements

CSC2E01: ARTIFICIAL INTELLIGENCE

Course Number: 6
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the basics of Artificial Intelligence and its applications.

CO2: Formalize the problem as a state space or graph

CO3: Apply search and game based techniques to solve state space problem assigned with heuristics.

CO4: Understand basic issues of knowledge representation techniques

CO5: Compare various machine learning methods.

CO6: Understand the basics of Genetic Algorithm and ANN.

Unit I :

Introduction: Artificial Intelligence- problems, scope and applications, Problem space and search- Production system- characteristics- the predicate calculus, Inference rules, Structures and strategies for state space search, strategies for space search, using state space to represent reasoning with the predicate calculus.

Unit II :

Heuristics Search: Control and implementation of state space search, Generate and test, Hill climbing, Depth first search, Breadth first search, Best-first search, A*, Problem Reduction, AO*, Constraint Satisfaction, Means-ends analysis, Heuristic in games, Complexity issues.

Unit III :

Game Playing - The Minimax search procedure, adding Alpha-beta cutoffs, Additional refinement, Iterative deepening, Planning system and its components, Understanding, Understanding as constrained satisfaction.

Unit IV :

Knowledge representation issues, representation and mappings, Representing simple facts in logic, Representing instances and ISA relationships, Computable functions and Predicates, Resolution, conversion to

clausal form, Unification algorithm, Natural deduction. Knowledge representation using rules, logic programming, forward versus backward reasoning, Symbolic reasoning under uncertainty- Nonmonotonic reasoning, Slot and filler structures: Semantic nets, frames, conceptual dependency, scripts.

Unit V :

Learning from Examples- Forms of learning-Supervised learning, Learning Decision Trees, Theory of Learning, Linear Regression and Classification, Deep Learning, Feed Forward Networks, Gradients and Learning, Back-Propagation, Computation graph for Deep Learning, Convolutional Networks, Recurrent Neural Networks, Unsupervised learning and Transfer learning, Reinforcement learning, Learning from rewards, Passive and Active Reinforcement Learning, Applications.

References:

1. E. Rich, K. Knight and S.B.Nair, Artificial Intelligence, 3rd Edn. TMGH, New Delhi, 2009.
2. G.F. Luger and W.A Stubblefield, Artificial Intelligence - Structures and Strategies for complex problem solving, Addison-Wesley-1998.
3. P.H Winston - Artificial Intelligence, Addison-Wesley-1992.
4. Nils J. Nilsson ,Artificial Intelligence , A New Synthesis, Morgan Kauf 2000.
5. W.F. Clocksin and C.S. Mellish, Springer Verlag, Programming in Prolog 2003
6. Dan W.Patterson , Introduction to Artificial Intelligence and Expert Systems, Prentice Hall, 1990

CSC2E02-Pattern Recognition

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the fundamental concepts of pattern recognition and machine learning.

CO2 : Recognize the principles of Bayesian parameter estimation.

CO3: Apply pattern recognition methods for pre-processing, feature extraction, and feature Selection to multivariate data.

CO4: Understand basics of Bayes classifier and linear discriminant analysis.

CO5: Describe the concepts of Syntactic methods - stochastic search- Boltzmann learning - Nonmetric methods- decision trees-CART

CO6: Understand supervised and unsupervised classification methods to detect and characterize patterns in real-world data.

Unit I :

Introduction - introduction to statistical - syntactic and descriptive approaches - features and feature extraction - learning - Bayes Decision theory - introduction - continuous case - 2- category classification - minimum error rate classification - classifiers - discriminant functions - and decision surfaces - error probabilities and integrals - normal density - discriminant functions for normal density.

Unit II :

Parameter estimation and supervised learning - maximum likelihood estimation - the Bayes classifier - learning the mean of a normal density - general Bayesian learning - nonparametric technic - density estimation - parzen windows - k-nearest neighbour estimation - estimation of posterior probabilities - nearest-neighbour rule - k-nearest neighbour rule.

Unit III :

Linear discriminant functions - linear discriminant functions and decision surfaces - generalized linear discriminant functions - 2-category linearly separable case - non-separable behaviour -linear programming algorithms, support vector machines- multilayer neural networks - feed

forward operation and classification, back propagation algorithm, error surface, back propagation as feature mapping.

Unit IV :

Syntactic methods - stochastic search- Boltzmann learning - Nonmetric methods- decision trees-CART - other tree methods, grammatical methods, grammatical inference.

Unit V :

Unsupervised learning and clustering - mixture densities and identifiability, maximum likelihood estimates, applications to normal mixtures, unsupervised Bayesian learning, data description and clustering.

References:

1. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, Second edition, 2006
2. Gonzalez R.C. and Thomson M.G., Syntactic Pattern Recognition - An Introduction, Addison Wesley.
3. Fu K.S., Syntactic Pattern Recognition And Applications, Prentice Hall, Eaglewood cliffs
4. Rajan Shinghal, Pattern Recognition: Techniques and Applications, Oxford University Press, 2008.

CSC2E03: Digital Signal Processing

Course Number: 6
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the modern digital signal processing algorithms and applications.

CO2: Analysis of discrete time signals.

CO3: Understand the use of digital systems in real time applications

CO4: Apply the algorithms for a wide area of recent applications.

CO5: To describe signals mathematically and understand how to perform mathematical operations on signals.

CO6: Knowledge about Digital filters

Unit I :

Signals and Signal Processing - Characterization and classification of Signals, Typical signal processing operations, Typical Signal Processing Applications, Advantage of Digital Signal Processing.

Unit II :

Classification of signals –Introduction to vector space - The concept of frequency in continuous and discrete time signals -Sampling of analog signals – Sampling theorem – Quantization and Coding – Digital to analog conversion .

Unit III :

Time Domain Representation of signals and systems - Discrete time signals, Operations on sequences, Discrete time Systems, Linear Time invariant Discrete Time Systems-convolution sum – correlation of discrete time signals .

Unit IV :

Review of Z-transforms, Properties of Z-transform, Rational Ztransforms, Inversion of Z- transforms, stability and causality.

Unit V :

Frequency Analysis of Signals - Frequency Analysis of Continuous Time Signals, Frequency Analysis of Discrete Time Signals, Fourier Transform of discrete time signals –Discrete Fourier Transform (DFT). FFT(Qualitative idea only)-Wavelet Transform - FIR and IIR Filters.

References:

1. Proakis, John G. and Dimitris G. Manolakis. Digital signal processing: principles, algorithms and applications. Pearson Education India, 2001.
2. Roberts, Michael J. Signals and systems: analysis using transform methods and MATLAB. McGraw-Hill Higher Education, 2011.
3. Oppenheim, Alan V., and Ronald W. Schafer. Digital Signal Processing [by] Alan V. Oppenheim [and] Ronald W. Schafer.Prentice-Hall, 1975.
4. Antoniou, Andreas. Digital signal processing.McGraw-Hill, 2016.
5. Rabiner, Lawrence R., Bernard Gold, and C. K. Yuen. Theory and application of digital signal processing. Prentice-Hall, 2007.

CSC2E04-Cryptography and Network Security

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the fundamentals of cryptography.

CO2: Interpret data integrity, authentication, digital signatures.

CO3: Analyze different network security applications

CO4: Familiarize standard algorithms that provide confidentiality, integrity and authenticity.

CO5: Develop knowledge about network security technologies.

CO6: To understand various protocols for network security to protect against the threats in the networks.

Unit I :

Computer security concepts - challenges - security attacks - security services - security mechanisms - a model for network security. Cryptography - symmetric encryption principles - cryptography - cryptanalysis - Feistal Cipher structure. Symmetric block encryption algorithms - DES - Triple DES - AES - random and pseudorandom numbers - stream cipher and RC4 - cipher block modes of operation.

Unit II :

Message authentication - approaches - MAC - one way Hash function - secure Hash functions - Message Authentication Codes. Public key cryptography principles - algorithms - digital Signatures.

Unit III :

Network security applications - symmetric key distributions using symmetric encryption - Kerberos version 4 - key distributions using asymmetric encryption - X.509 certificates -public key infrastructure - federated identity management.

Unit IV :

Transport level security - web security considerations - secure socket layer and transport layer security - SSL architecture - SSL record protocol - change cipher spec protocol - handshake protocol. Transport

layer security - HTTPS - SSH. IP Security - overview - policy - encapsulating security payload - combining security associations - internet key exchange.

Unit V :

Intruders - intruders, intrusion detection, password management. Malicious software - types, viruses, countermeasures, worms, DDoS. Firewalls - need - characteristics, types, firewall basing, location and configuration - DMZ networks, VPN - distributed firewalls.

References:

1. William Stallings, Network Security Essentials Applications and Standards, 4th Edition, Pearson India, ISBN: 8131761754.
2. William Stallings, Cryptography and Network Security : Principles and Practice, 6th Edition, Pearson India, ISBN: 9332518777.
3. Atul Kahate, Cryptography and Network Security, 3rd Edition, Tata McGraw-Hill Publishing, ISBN: 9789332900929.
4. Eric Maiwald, Fundamental of Network Security, 1st Edition, Tata McGraw - Hill Education, 0071070931.
5. Charlie Kaufman, Radia Perlman and Mike Speciner, Network Security: Private Communication in Public World, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120322134.

CSC2E05: Simulation and Modeling

Course Number: 6
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Illustrate concept, principles of mathematical modeling and discrete system simulation.

CO2: Analyse various techniques for evaluation of simulation experiments

CO3: Employ the concept of simulation of queueing systems and stochastic network

CO4: Generate random numbers and random variates using different techniques.

CO5: Develop a simulation model using heuristic methods.

CO6: Analysis of Simulation models using input analyzer, and output analyzer

Unit I :

Introduction - systems and models - computer simulation and its applications -continuous system simulation - modeling continuous systems - simulation of continuous systems - discrete system simulation - methodology - event scheduling and process interaction approaches - random number generation.

Unit II :

Testing of randomness - generation of stochastic variates - random samples from continuous distributions - uniform distribution - exponential distribution m-Erlang distribution - gamma distribution - normal distribution - beta distribution - random samples from discrete distributions Bernoulli - discrete uniform -binomial - geometric and poisson.

Unit III :

Evaluation of simulation experiments - verification and validation of simulation experiments - statistical reliability in evaluating simulation experiments -confidence intervals for terminating simulation runs - simulation languages -programming considerations - general features of GPSS-SIM SCRIPT and SIMULA.

Unit IV :

Simulation of queueing systems - parameters of queue - formulation of queueing problems - generation of arrival pattern - generation of service patterns -Simulation of single server queues - simulation of multi-server queues -simulation of random queues.

Unit V :

Simulation of stochastic network - simulation of PERT network - definition of network diagrams forward pass computation - simulation of forward pass -backward pass computations - simulation of backward pass - determination of float and slack times determination of critical path - simulation of complete network - merits of simulation of stochastic networks.

References:

1. C. Deo N., System Simulation And Digital Computer, Prentice Hall of India.
2. Gordan G., System Simulation, Prentice Hall of India.
3. Law A.M. & Ketton W.D., Simulation Modelling and Analysis, McGraw Hill.

CSC2A02: Term Paper

Course Number: 7
Number of Credits: 2
Course Outcomes:

CO1: Apply critical thinking and analytical ability in problem solving.

CO2: Practice the research methodology for literature survey.

CO3: Apply foundational research skills to address a research problem.

CO4: Innovate, experiment and analyze research findings.

CO5: Demonstrate capacity to lead and manage change through collaborative environment.

CO6: Practice the process of presenting the research work through seminars/conferences and scientific publishing.

A term paper is introduced in the second semester with the following objectives.

- To familiarize the student to the techniques of literature survey and to conduct a study/analysis result in a critical review of the recent research work/technology innovations related to Computer Science.
- To acquaint with the process of presenting the work through seminars and technical reports.

The student is expected to do an extensive literature survey and analysis in a research/innovations/technology area related to Computer Science, under the supervision of a faculty member from the Department. The student has to choose an area/topic for his/her work after due consultation and approval from the guide (students can refer articles from ACM/IEEE/INFLIBNET Journals/Conference Proceedings and/or equivalent documents, standard textbooks and web based material, approved by the supervisor)

The study should preferably result in a critical review of the recent research works/technology advancements/ innovations/ algorithms/theoretical contributions in the form of theorems and proofs/new methods of proof/new techniques or heuristics with analytical studies/implementations and analysis of results.

The student should give a seminar on his/her work, during the semester, and submit a technical report.

Evaluation: For the evaluation of the term paper, an evaluation committee is to be constituted. One faculty is to be designated as the Course Coordinator for this course. Committee is to be constituted by the Head of the Department (HOD) and (s)he shall be the Chairperson of the committee. In addition to the HOD, the Course Coordinator, the faculty guiding a particular student will also be a member of the committee. The Coordinator has to set the schedule for presentation and submission of the reports. While calculating the final score, 25% weight is to be given for the scores awarded by the guide to the student and the rest 75% weight is to be given for the average of the scores awarded to the student by remaining committee members.

A tentative list of the components for evaluation of Term Paper is as shown below. Evaluation committee can decide about the actual composition of the components and scores to be awarded for each component.

Components:

- Relevance of the Topic, Statement of Objectives, Correctness
- Quality of Literature Survey / Analysis of results
- Methodology / Tools adopted
- Quality of Implementation/Simulation /Testing
- Identification of Future Work
- Quality of the Term Paper Report
- Quality of Presentation
- Publications out of the Term Paper

Third Semester

CSC3C12: Principles of Compiler Design

Course Number: 1
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Model the process of translating a high-level language to an executable code.

CO2: Describe how different phases of a compiler work.

CO3: Experiment top down and bottom up parsing algorithms.

CO4: Apply flow graph for the intermediate codes.

CO5: Apply optimization techniques to have a better code for code generation.

CO6: Understand the basics of Storage organization and Storage allocation strategies.

Unit I :

Introduction to compiling : Compilers - Analysis of the source program - Phases of a compiler - Cousins of the Compiler - Grouping of Phases - Compiler construction tools - Lexical Analysis - Role of Lexical Analyzer - Input Buffering - Specification of Tokens.

Unit II :

Syntax analysis : Role of the parser - Writing Grammars - Context-Free Grammars - Top Down parsing - Recursive Descent Parsing - Predictive Parsing - Bottom-up parsing - Shift Reduce Parsing - Operator Precedent Parsing - LR Parsers - SLR Parser - Canonical LR Parser - LALR Parser.

Unit III :

Intermediate code generation: Intermediate languages - Declarations - Assignment Statements - Boolean Expressions - Case Statements - Back patching - Procedure calls.

Unit IV :

Code generation: Issues in the design of code generator - The target machine - Runtime Storage management - Basic Blocks and Flow Graphs - Next-use Information - A simple Code generator - DAG representation of Basic Blocks - Peephole Optimization.

Unit V :

Code optimization and run time environments : Introduction - Principal Sources of Optimization - Optimization of basic Blocks - Introduction to Global Data Flow Analysis - Runtime Environments - Source Language issues - Storage Organization - Storage Allocation strategies - Access to non-local names - Parameter Passing.

References:

1. V Aho A, Ravi Sethi, D Ullman J, Compilers Principles, Techniques and Tools, 2nd Edition, Pearson Education Singapore Pte Ltd.
2. K. V. N. Sunitha, Compiler Construction, Pearson.
3. W Appel and Andrew, Modern Compiler Implementation in C, 1st Edition, Cambridge University Press.
4. Allen I Holub, Compiler Design in C, 1st Edition, PHI Learning Pvt Ltd.
5. Tremblay and Sorenson, The Theory and Practice of Compiler Writing, 1st Edition, BSP Books Pvt Ltd.
6. Torben Egidius Mogensen, Basics of Compiler Design, Department of Computer Science, University of Copenhagen (Online Edition).

CSC3C13: Data Communication and Networking

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 3
Course Outcomes:

CO1: Illustrate the basics of data and signal.

CO2: Familiarize with OSI reference model.

CO3: Identify protocols at different layers.

CO4: Understand an overview on various LAN architectures.

CO5: Practice the concepts of computer network security.

CO6: Understand errors in data communications and their correction.

Unit I :

Data and signals -Analog and Digital signals, Line configuration, Topology, Transmission mode, Guided and Unguided media, Parallel and Serial data transmission, Layered approaches-OSI and TCP/IP. Multiplexing. Transmission impairments- Distortion, Attenuation, Noise. Analog modulation- AM, PM, FM. Sampling theorem, Analog pulse modulation, Digital pulse modulation. ASK, FSK, PSK and PSK. Digital data transmission and Interface standards. DTE and DCE. Modems.

Unit II :

Data Link Layers – Error Detection and Correction. Data Link Control – line Discipline, flow control, error control .Data link Protocol-synchronous and asynchronous protocol, character oriented and Bit Oriented protocol. Multi Access Protocols- ALOHA and CSMA. Switching. Networking and Internetworking devices-Repeaters, Bridges, Routers and Gateways. LANs: Ethernet, token bus, token ring. Bluetooth-architecture.

Unit III :

Network Layer: address and protocol- ARP, IP, ICMP, IGMP.IPv4 and IPv6.Subnetting, Routing Algorithms – Shortest path, Flooding, Distance Vector Routing, Link state Routing, Congestion Control Algorithms. Internetworking: Tunneling, Internetwork Routing, Fragmentation, IPv4 Vs IPv6Protocol, IP Addresses, Internet Control Protocols–ICMP, ARP, RARP, DHCP.

Unit IV :

Transport Layer- Design Issues, Connection Management, Transmission Control Protocol (TCP), User Datagram Protocol (UDP). Application layer: Protocols, DNS, Telnet, www and HTTP.

Unit V :

Computer Network Security- Introduction, Need for security, Principles of Security, Types of Attacks. Symmetric and Public key algorithms. Authentication. Integrity,Key Distribution and Certification.

References:

1. Behrouz A.Forouzan,Data Communicationsand Networking,Fourth Edition,McGrawHill 2001
2. AndrewS.Tanenbaum,Computer Networks,FourthEdition,Prentice-Hall,2003
3. WilliamStallings,DataandComputer Communication, EighthEdition,Prentice-Hall,2007
4. J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach Featuring Internet, Sixth Edition, Perason Education.

CSC3C14 - Web Technology

Course Number: 3
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Practice the concepts of Web programming including HTML5 futures and tag set.

CO2: Practice server side and client side scripting.

CO3: Develop applications using JSP scripting elements by Linking to external files and Accessing databases from JSP.

CO4: Develop applications with advanced features of PHP by establishing connection with MySQL databases.

CO5: Analyze server side programming with Python.

CO6: Understand the difference between the HTML, PHP and XML documents.

Unit I :

Introduction to Web programming - Introduction to SGML features - HTML, XHTML, DHTML, XML - HTML Vs XML - Creating XML documents - Parsing an XML document - Writing well formed documents - Organizing elements with namespaces - Defining elements in a DTD - Declaring elements and attributes in a DTD. Familiarization of HTML5 futures and tag set.

Unit II :

Introduction to scripting - server side and client side scripting. CGI/Perl: Creating link to a CGI Script - Using a link to send data to a CGI Script - parsing data sent to a Perl CGI script - Using CGI script to process form data - Using scalar variables in Perl - Using variables in Perl - Using arithmetic operators in Perl - arrays, list, hashes, strings, pattern and regular expressions, text processing, subroutines - Associating a form with a script-

Unit III :

Java Server Pages - JSP scripting elements - Linking to external files - JSP declarations - JSP Expressions - JSP Scriplets - Processing client

requests- Accessing a database from JSP. Advantages of JSP comparing with other language.

Unit IV :

PHP: Advanced features of PHP, Defining PHP variables - variable types - operators - control flow constructs in PHP- Function, Creating a Function, Function Libraries, Arrays, strings and Regular Expressions, GET and POST methods - Establishing connection with MySQL database - managing system data - parsing data between pages

Unit V :

Python: Introduction to Python language, Advantages of Python in comparison with other Languages, Data types, control structures, advanced data structures, I/O, defining classes, data collections, functions and modules (math module), packages, exception handling, standard library, internet programming with python.

References:

1. Robert W. Sebesta, Programming with World Wide Web, 4th edition, Pearson Education, 2009.
2. XueBal et. al, The Web Warrior Guide to Web programming, Thomson Learning.
3. Chris Bates, Web Programming: Building Internet Applications, 3rd ed, Wiley Academic Catalog.
4. H.M. Deitel, P.J. Deitel and A.B. Goldberg, Internet and World Wide Web: How to Program, rd edition, Pearson Education.
5. Steven Holzner, PHP The complete Reference,1stEdition,McGraw-Hill, 2007.
6. Philip Hanna, JSP The complete Reference, 2nd Edition, McGraw-Hill, 2002.
7. Paul Barry, Head First Python, 1st Edition, O'Reilly Media, 2010
8. Scott Guelicb, ShishirGundavaram and Gunther Birznieks, CGI Programming with Perl, 2ndEdition,O'Reilly Media, 2000.
9. How to Think Like a Computer Scientist: Learning with Python, Allen Downey , Jeffrey Elkner , Chris Meyers, http://www.greenteapress.com/thinkpython/thinkpython.pdf

CSC3E06: Digital Image Processing

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Familiarize image sampling and quantization techniques.

CO2: Examine image transformation techniques viz. Fourier transform, Walsh Hadamard, DCT, and Hotelling transform.

CO3: Examine image enhancement techniques - histogram processing and various image filters viz.laplacian filter, smoothing and sharpening filters, spatial filters, and homomorphic filters.

CO4: Understand Image restoration process, noise models and inverse filtering Techniques.

CO5: Understand Edge detection, region based segmentation and boundary representation

CO6: Familiarise various Image compression models - information theoretic perspective.

Unit I :

Introduction - digital image representation - fundamental steps in image processing - elements of digital image processing systems - digital image fundamentals - elements of visual perception - a simple image model - sampling and quantization - basic relationship between pixels - image geometry

Unit II :

Image transforms - introduction to Fourier transform - discrete Fourier transform (DFT) - properties DFT- other separable image transforms - Walsh, Hadamard and Discrete Cosine transforms. Hotelling transform.

Unit III :

Image enhancement - basic grey level transformation - histogram equalization - image subtraction - Image averaging - spatial filtering - smoothing, sharpening filters - Laplacian filters. Enhancement in the frequency domain - frequency domain filters - smoothing, sharpening filters - homomorphic filtering.

Unit IV :

Image restoration - model of Image degradation/restoration process - noise models - inverse filtering - least mean square filtering - constrained least mean square filtering. Edge detection - thresholding - region based segmentation - Boundary representation

Unit V :

Image compression - fundamental concepts of image compression - compression models - information theoretic perspective. Lossless compression - Huffman coding - arithmetic coding - bit plane coding - run length coding. Lossy compression - transform coding - Image compression standards.

References:

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing - 3rd ed., Prentice Hall of India, New Delhi, 2008
2. B. Chanda and D.D. Majumder, Digital Image Processing and Analysis, PHI
3. A.K. Jain, Fundamentals of Digital Image Processing, PHI
4. W.K. Pratt, Digital Image Processing, John Wiley, 2006
5. M. Sonka, V. Hlavac and R. Boyle, Image Processing Analysis and Machine Vision, Brooks/colic, Thompson Learning, 1999.

CSC3E07 - Digital Speech Processing

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the significance of digital speech processing and its applications

CO2: Familiarize speech production mechanism and corresponding physical models

CO3: Analyze linear prediction methods for modeling of speech signal

CO4: Describe homomorphic speech processing for estimation of excitation and vocal tract model

CO5: Analyze quantization effects of model coefficients on its accuracy

CO6 Apply methods for recognition of vowels and speaker identity.

Unit I :

Basic Concepts: Speech Fundamentals: Articulatory Phonetics – Production and Classification of Speech Sounds; Acoustic Phonetics – acoustics of speech production; Review of Digital Signal Processing concepts; Short-Time Fourier Transform, Filter-Bank and LPC Methods.

Unit II :

Speech Analysis: Features, Feature Extraction and Pattern Comparison Techniques: Speech distortion measures – mathematical and perceptual – Log Spectral Distance, Cepstral Distances, Weighted Cepstral Distances and Filtering, Likelihood Distortions.

Unit III :

Spectral Distortion using a Warped Frequency Scale, LPC, PLP and MFCC Coefficients, Time Alignment and Normalization – Dynamic Time Warping, Multiple Time – Alignment Paths.

Unit IV :

Speech Modeling: Hidden Markov Models: Markov Processes, HMMs – Evaluation, Optimal State Sequence – Viterbi Search, Baum-Welch Parameter Re-estimation, Implementation issues.

Unit V :

Speech Recognition: Large Vocabulary Continuous Speech Recognition: Architecture of a large vocabulary continuous speech recognition system – acoustics and language models – ngrams, context dependent sub-word units; Applications and present status.

References:

1. Rabiner, Lawrence R., Biing-Hwang Juang, and Janet C. Rutledge. Fundamentals of speech recognition.Vol. 14. Englewood Cliffs: PTR Prentice Hall, 1993.
2. L R Rabiner and Schafer ,Digital processing of speech signals, Prentice hall. 1978.
3. Proakis, John G. Digital signal processing: principles, algorithms and applications. Pearson Education India, 2001.
4. AJurafsky, Dan. Speech & language processing. Pearson Education India, 2000.

CSC3E08-Natural Language Processing

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the fundamental concepts of Natural Language Processing.

CO2: Design algorithms for solving NLP based problems.

CO3: Develop useful systems for language processing and related tasks involving text processing.

CO4: Understand how to employ literary-historical NLP-based analytic techniques like stylometry, topic modeling, synsetting and named entity recognition in personal research.

CO5: Describe the methods used for part of speech tagging and named entity recognition.

CO6: Design solutions for solving text or information extraction, summarization and classification problems.

Unit I :

Introduction - Models -and Algorithms - The Turing Test -Regular Expressions Basic Regular Expression Patterns -Finite State Automata - Regular Languages and FSAs - Morphology - Inflectional Morphology - Derivational Morphology -Finite-State Morphological Parsing - Combining an FST Lexicon and Rules -Porter Stemmer.

Unit II :

N-grams Models of Syntax - Counting Words - Unsmoothed N-grams - Smoothing- Backoff -Deleted Interpolation - Entropy - English Word Classes - Tagsets for English - Part of Speech Tagging -Rule-Based Part of Speech Tagging - Stochastic Part of Speech Tagging - Transformation-Based Tagging.

Unit III :

Context Free Grammars for English Syntax- Context-Free Rules and Trees - Sentence- Level Constructions - Agreement - Sub Categorization - Parsing - Top-down - Earley Parsing - Feature Structures - Probabilistic Context-Free Grammars.

Unit IV :

Representing Meaning - Meaning Structure of Language - First Order Predicate Calculus-Representing Linguistically Relevant Concepts -Syntax-Driven Semantic Analysis - Semantic Attachments - Syntax-Driven Analyzer - Robust Analysis - Lexemes and Their Senses - Internal Structure - Word Sense Disambiguation -Information Retrieval.

Unit V :

Discourse -Reference Resolution - Text Coherence -Discourse Structure - Dialog and Conversational Agents - Dialog Acts - Interpretation - Coherence - Conversational Agents - Language Generation - Architecture - Surface Realizations - Discourse Planning - Machine Translation - Transfer Metaphor - Interlingua - Statistical Approaches.

References:

1. D.Jurafsky and J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,2nd Edition, Prentice Hall, 2008.
2. C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing ", MIT Press, 1999.
3. James Allen. Natural Language Understanding, Addison Wesley, 1994.

CSC3E09-Bioinformatics

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the basic concepts of Bioinformatics and its significance in Biological data analysis.

CO2: Understand the popular genomic and proteomic databases and to impart knowledge in processing and analysing genomic data.

CO3: Identify steps in sequence alignment.

CO4: Conceptualise the advanced topics in Bioinformatics.

CO5: Understand Basic concepts of phylogeny.

CO6: Develop Visualization of protein structures using Rasmol or Rastop.

Unit I :

Cells-Prokaryotes and Eukaryotes-DNA double helix- central dogma - RNA, Amino acids, Proteins -string representations- different levels of protein structures-DNA cloning- A brief introduction to different mappings techniques of genomes- genome sequencing methods-DNA micro arrays - Human Genome Project-A glossary of biological terms.

Unit II :

Scope of bioinformatics-Genomics and Proteomics- Problems in bioinformatics - sequence alignment, phylogeny, gene finding, microarray analysis, Homology and evolutionary relationships; Homology analysis and function of an entire gene or of segments within it, secondary structure prediction, protein structure prediction, comparative genomics and drug design.

Unit III :

Data management, Data life cycle, An introduction to the major resources at NCBI, EBI and ExPASy - Nucleic acid sequence databases: GenBank, EMBL, DDBJ - Protein sequence databases: SWISS-PROT, TrEMBL, PIR-PSD - Genome Databases at NCBI, EBI, TIGR, SANGER - How to access these databases and to make use of the tools available. Various file formats for bio-molecular sequences like genbank and fasta,

the concept of profiles- The derived databases- Prosite, Pfam, PRINTS, CATH, SCOP.

Unit IV :

Basic concepts of sequence similarity, identity and homology, definitions of homologues, orthologues, paralogues. Scoring matrices: basic concept of a scoring matrix, PAM and BLOSUM matrices, differences between distance and similarity matrix, Pairwise sequence alignments: basic concepts of sequence alignment, Needleman and Wuncsh, Smith and Waterman algorithms for pairwise alignments, BLAST and FASTA and their versions, Multiple sequence alignments (MSA): the need for MSA, basic concepts of various approaches for MSA (e.g. progressive, hierarchical etc.), Algorithm of CLUSTALW.

Unit V :

Phylogeny: Basic concepts of phylogeny; molecular evolution; Definition and description of phylogenetic trees, Phylogenetic analysis algorithms - Maximum Parsimony, UPGMA and Neighbour-Joining, Gene Finding: The six reading frames-Computational gene finding in prokaryotes and eukaryotes Basic signals - start and stop codons, promoters etc- important coding measures- Regular expressions- Introduction to Hidden Markov models- Introduction to genomic signal processing Molecular visualization: Visualization of protein structures using Rasmol or Rastop

References:

1. Hooman H. Rashidi and Lukas K.Buehler, Bioinformatics Basics. Applications in Biological Science and Medicine, CAC Press 2000.
2. Dan Gusfiled, Algorithms on Strings Trees and Sequences, Cambridge University Press 1997.
3. P. Baldi. S. Brunak, Bioinformatics: A Machine Learning Approach, MIT Press, 1988.
4. Harshawardhan P.Bal, Bioinformatics - Principles and Applications - Tata MeGraw Hill

CSC3E10 - Computer Vision

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand fundamental image processing techniques required for computer vision

CO2: Familiarize Image formation process and shape analysis

CO3: Identify features from Images and do analysis of Images

CO4: Describe 3D model from images

CO5: Model applications using computer vision techniques

CO6: Understand video processing, motion computation and 3D vision and geometry

Unit I :

Introduction : Image Processing, Computer Vision and Computer Graphics , What is Computer Vision - Low-level, Mid-level, High-level , Overview of Diverse Computer Vision Applications: Document Image Analysis, Biometrics, Object Recognition, Tracking, Medical Image Analysis, Content-Based Image Retrieval, Video Data Processing, Multimedia, Virtual Reality and Augmented Reality.

Unit II :

Image Formation Models : Monocular imaging system , Radiosity: The 'Physics' of Image Formation, Radiance, Irradiance, BRDF, color etc, Orthographic & Perspective Projection,• Camera model and Camera calibration, Binocular imaging systems, Multiple views geometry, Structure determination, shape from shading , Photometric Stereo, Depth from Defocus , Construction of 3D model from images.

Unit III :

Image Processing and Feature Extraction: Image preprocessing, Image representations (continuous and discrete) , Edge detection .Motion Estimation : Regularization theory , Optical computation , Stereo Vision , Motion estimation , Structure from motion .

Unit IV :

Shape Representation and Segmentation : Contour based representation,

Region based representation, Deformable curves and surfaces , Snakes and active contours, Level set representations , Fourier and wavelet descriptors , Medial representations , Multi-resolution analysis.

Unit V :

Object recognition : Hough transforms and other simple object recognition methods, Shape correspondence and shape matching , Principal component analysis , Shape priors for recognition. Image Understanding : Pattern recognition methods, HMM, GMM and EM. Applications: Photo album ,Face detection , Face recognition ,Eigenfaces ,Active appearance and 3D shape models of faces

References:

1. Introductory Techniques for 3D Computer Vision, by E. Trucco and A. Verri, Publisher: Prentice Hall.
2. Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall Robot Vision, by B. K. P. Horn, McGraw-Hill.
3. Image Processing, Analysis, and Machine Vision. Sonka, Hlavac, and Boyle. Thomson
4. R. C. Gonzalez, R. E. Woods. Digital Image Processing. Addison Wesley Longman, Inc., 1992.
5. D. H. Ballard, C. M. Brown. Computer Vision.Prentice-Hall, Englewood Cliffs, 1982.
6. Richard Szeliski, Computer Vision: Algorithms and Applications (CVAA). Springer, 2010
7. Mark Nixon and Alberto S. Aquado, Feature Extraction & Image Processing for Computer Vision, Third Edition, Academic Press, 2012
8. E. R. Davies, Computer & Machine Vision, Fourth Edition, Academic Press, 2012
9. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012

CSC3E11 - Data Analytics Using Python

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the overview of data analytics in the context of machine learning and artificial intelligence

CO2: Experiment exploration through python by familiarising packages necessary to perform various data analytics tasks.

CO3: Design various machine learning algorithms

CO4: Apply data preparations and analysis to solve real word problems.

CO5: Familiarize Python Packages for Basic Data Analytics - numpy, pandas, matplotlib.

CO6: Design Exploratory and Explanatory data analysis using Python packages

Unit I :

Overview of data science and analytics, Need of data science, Role and skills required for a data scientist, Technologies used in data analytics, Data analytics tools. Overview of artificial intelligence and machine learning. Division of machine learning tasks - supervised and unsupervised. Learning, predicting and classifying using machine learning.

Unit II :

Python: Introduction to Python language, Advantages of Python in comparison with other Languages, Characteristics of python, suitability of python in data manipulation and analysis, Data types, control structures, advanced data structures, I/O, defining classes, data collections, functions and modules, packages, exception handling and standard library in python. Computation capability of python. Powerfulness of libraries and tools of python in data manipulation, visualisation and data plotting like numpy, pandas, matplotlib, seaborn, scikit-learn.

Unit III :

Overview of machine learning algorithms, suitability of machine learning algorithms in data science problems and analytics, Application of machine learning in feature selection, Generalization of data using machine

learning, Building predictive model by training with data and model evaluation, Need of dimensionality reduction of data.

Unit IV :

Regression analysis, techniques and its implementation using python, Regularisation techniques, Clustering, Cluster validation. Classification, algorithms for classification, Accuracy of classification techniques, Confusion matrix.

Unit V :

Formation of a data set, development of training data, testing data, and validation data, cross validation. Analysis of a problem and its relevant dataset - data analysis, modeling and evaluation of this case including the processes of Regression, clustering, feature selection, dimensionality reduction, classification, cross validation etc -implementation in Python.

References:

1. Jesus Rogel-Salazar,Data Science and Analytics with Python,CRCPress,1st Edition,2017.
2. Andreas C Muller, Sarah Guido, Introduction to machine learning with Python,O'Reilly,1st edition,2016.
3. Jake VanderPlas, Python Data Science Handbook - Essential Tools for Working with Data,O'Reilly,1st edition,2016.
4. Joel Grus, Data Science from Scratch: First Principles with Python, O'Reilly, 1st Edition,2015

CSC3E12 - Data Mining

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Identify the scope and necessity of Data Mining and Warehousing for the society

CO2: Understand data quality and methods and techniques for preprocessing of data.

CO3: Analyze the patterns that can be discovered by classification and prediction.

CO4: Understand the data mining techniques based on cluster analysis

CO5: Identify complex data types with respect to graph mining, spatial and web mining.

CO6: Become proficient in the use of basic data mining tools..

Unit I :

Data warehouse - definition - operational database systems Vs data warehouses - multidimensional model - from tables and spreadsheets to Data Cubes - schemas for multidimensional databases - measures - concept hierarchies - OLAP operations in the multidimensional data model - data warehouse architecture.

Unit II :

Data mining - introduction - definition - data mining functionalities - major issues in data mining - data preprocessing - data cleaning - data integration and transformation - data reduction - data discretization and concept hierarchy generation. Association rule mining - efficient and scalable frequent item set mining methods - mining various kinds of association rules - association mining to correlation analysis - constraint-based association mining.

Unit III :

Classification and prediction - issues regarding classification and prediction - classification by decision tree introduction - Bayesian classification - rule based classification - classification by back propagation - support

vector machines - associative classification - lazy learners - other classification methods - prediction - accuracy and error measures - evaluating the accuracy of a classifier or predictor - ensemble methods - model section.

Unit IV :

Cluster analysis - types of data in cluster analysis - a categorization of major clustering methods - partitioning methods - hierarchical methods - density-based methods - grid-based methods - model-based clustering methods - clustering high dimensional data - constraint-based cluster analysis - outlier analysis.

Unit V :

Graph mining - mining object, spatial, multimedia, text and web data - multidimensional analysis and descriptive mining of complex data objects - spatial data mining - multimedia data mining - text mining - mining the World Wide Web.

References:

1. Jain Pei, Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, 3rd Edition, Elsevier.
2. Alex Berson and Stephen J. Smith, Data Warehousing, Data Mining and OLAP, Computing Mcgraw-Hill.
3. K.P. Soman, Shyam Diwakar and V. Ajay, Insight into Data mining Theory and Practice, 1st Edition, Prentice Hall of India.

CSC3E13 - Information Retrieval Systems

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Practice the fundamentals of information retrieval systems and its taxonomy.

CO2: Illustrates the basic information retrieval systems and its evaluation strategies.

CO3: Understand the multimedia inputs and multimedia Information retrieval systems.

CO4: Modeling of information retrieval systems for search engines.

CO5: Analyze the performance of information retrieval using advanced techniques such as classification, clustering, and filtering over multimedia.

CO6: Demonstrate Information visualization technologies like Cognition and perception in the Internet or Web search engine.

Unit I :

Introduction: Information versus Data Retrieval, IR: Past, present, and future. Basic concepts: The retrieval process, logical view of documents. Modeling: A Taxonomy of IR models, ad-hoc retrieval and filtering. Classic IR models: Set theoretic, algebraic, probabilistic IR models, models for browsing.

Unit II :

Retrieval evaluation: Performance evaluation of IR: Recall and Precision, other measures, Reference Collections, such as TREC, CACM, and ISI data sets. Query Languages: Keyword based queries, single word queries, context queries, Boolean Queries, Query protocols, query operations.

Unit III :

Text and Multimedia Languages and properties, Metadata, Text formats, Markup languages, Multimedia data formats, Text Operations. Indexing and searching: Inverted files, Suffix trees, Suffix arrays, signature files, sequential searching, Pattern matching.

Unit IV :

Multimedia IR: Spatial access methods, Generic multimedia Indexing approach, Distance functions, feature extraction, Image features and distance functions. Searching the Web: Characterizing and measuring the Web.

Unit V :

Search Engines: Centralized and Distributed architectures, user Interfaces, Ranking, Crawling the Web, Web directories, Dynamic search and Software Agents. Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example - TREC results.

References:

1. Kowalski, Gerald, Mark T Maybury: Information Retrieval Systems: Theory and Implementation, Kluwer Academic Press, 1997.
2. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
3. Yates, Modern Information Retrieval, Pearson Education.
4. Robert Korfhage, Information Storage and Retieval, John Wiley and Sons.

CSC3E14 - Distributed Systems and Parallel Computing

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.

CO2: Understand different types of Distributed Systems,

CO3: Analyze the various design principles of parallel algorithms.

CO4: Learn the intricacies of parallel and distributed programming.

CO5: Develop and execute basic parallel and distributed applications using basic programming models and tools

CO6: Understand goals of parallelism, Parallelism and concurrency using multiple instruction streams.

Unit I :

Introduction to Distributed Systems:Goals of the Distributed Systems, Relation to parallel systems, synchronous versus asynchronous execution, design issues and challenges, Types of Distributed Systems, Distributed System Models, Hardware and software concepts related to distributed systems, middleware models.

Unit II :

Distributed Computing and Communication design principles: A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- Passing, Consensus, Conditional Actions, Critical Paths, Scalability, and cache coherence in multiprocessor systems, synchronization mechanism.

Unit III :

Introduction to Parallel Computing: The Idea of Parallelism, Power and potential of parallelism, Examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instruction streams.

Unit IV :

Parallel Architecture: Pipeline architecture, Array processor, Multi processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, Memory Issues : Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, Vector processing, GPU co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

Unit V :

Parallel and Distributed Programming Frameworks Overview of CUDA, OpenMP, POSIX Threads, Apache Hadoop (DFS), and current trends in parallel and distributed computing.

References:

1. Introduction to Parallel Computing (2nd Edition), Ananth Grama, Anshul Gupta, and George Karypis, Vipin Kumar, Addition Wesley
2. Parallel and Distributed Systems 2nd Edition, Arun Kulkarni, Nupur Prasasd Giri, Nikhilesh Joshi, Bhushan Jadhav, Wiley
3. Introduction To Parallel Programming, Steven Brawer, Academic Press
4. Introduction To Parallel Processing, M.Sasikumar, Dinesh Shikhare and P. Ravi Prakash, Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley
5. Distributed Operating Systems, A.S. Tanenbaum, Prentice Hall
6. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
7. Parallel Programming in C with MPI and OpenMP Michael J.Quinn, McGrawHill Higher Education.

CSC3O 01 - Introduction to Data Science

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the role of Data Science in business, industry and government.

CO2: Identify the roles and stages in a Data Science Projects and ethical issues.

CO3: Learn the processes, tools for the Analysis of structured and unstructured data.

CO4: Explore the importance of data visualization, storytelling with data.

CO5: Be able to draw accurate and useful conclusions from a data analysis.

CO6: Differentiate between ethical and unethical uses of data science.

Unit I :

Introduction to Data Science: Fundamentals of Data Science, Real World applications, Data Science vs Business Intelligence, Data Science vs Statistics, Roles and responsibilities of a Data Scientist, Software Engineering for Data Science, Data Science Challenges.

Unit II :

Data Analytics: Definition, Types, methodologies, CRISP-DM Methodology, SEMMA, BIG DATA LIFE CYCLE, SMAM, Challenges in Data-driven decision making, Data Science Process: Data Science methodology, Data Acquisition, Understanding, Data preparation, Modelling, Model Evaluation, Data Science Teams: Defining Data Team, Roles, Managing Data Team.

Unit III :

Data and Data Models: Types of Data and Data-sets, Data Quality, Data Models Formal modelling: General Framework, Associational Analyses, Prediction Analyses, Data wrangling and Feature Engineering: Data cleaning, Aggregation, Sampling, Handling Numeric Data: Discretization, Binarization, Normalization, Smoothening, Dealing with textual Data: Managing Categorical Attributes, Transforming Categorical

to Numerical Values, Encoding techniques, Feature Engineering: Feature Extraction (Dimensionality Reduction), Feature Construction, Feature Subset selection: Filter methods, Wrapper methods and Embedded methods, Feature Learning.

Unit IV :

Data visualization, Need for visualization, Exploratory vs Explanatory Analysis, Tables, Axis based Visualization and Statistical Plots, Data Visualization Design Process, Lessons in Data Visualization Design, Stories and Dashboards.

Unit V :

Ethics for Data Science, Bias and Fairness: Types of Bias, Identifying Bias, Evaluating Bias, Examples of misuse of Data, Doing Good Science, Five C's, Ethical guidelines for Data Scientist, Ethics of data scraping and storage.

References:

1. Cielen, D., Meysman, A.,Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Manning Publications, 2016.
2. Cole Nussbaumer Knaflic, Storytelling with Data, A data visualization guide for business professionals, by; Wiley, 2015.
3. P. Tan, M. Steinbach, AKarpatne, and V. Kumar, Introduction to Data Mining, 2nd EdPearson Education, 2018.
4. Roger D Peng and Elizabeth Matsui , The Art of Data Science by, Pelican Books, 2020,
5. Mike Loukides, Hilary Mason, DJ Patil, Ethics and Data Science, O'Reilly, 2018
6. Jake VanderPlas, Python Data Science Handbook: Essential tools for working with data, , O'Reilly, 2016.
7. W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython, 2nd Ed., O'Reilly, 2017.

CSC3C15 - PRACTICAL 3

Course Number: 6
Contact Hours per Week: 15 (10 Practical + 5 Tutorial)
Number of Credits: 3
Course Outcomes:

CO1: Develop programs related to the theory portions covered in CSC3C14 Web Technology.

Web Technology:

1. Create webpages using HTML, HTML5 including text, lists, Images and Videos, hyperlinks, tables and menus .
2. Design a data entry form using DHTML including java script validation.
3. Design a web page using XHTML including text, lists, data tables and anchor links.
4. Write an XML file describing some entity attributes. Write a document type definition(DTD) to validate the above XML file. Display the XML file contents in a coloured tabular format with suitable headings.
5. Programs to demonstrate the usage of regular expressions in perl.
6. Database programming with perl
7. Programming and web development using CGI-Perl.
8. Create a webpage using JSP and JDBC including operations of data storage and retrieval from database tables.
9. Develop dynamic database driven sites with PHP and MySQL.
10. Programming and web development using Python.

Fourth Semester

CSC4C16 - Software Engineering

Course Number: 1
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand software process models and web engineering concepts.

CO2: Understand software requirements and structured analysis of systems.

CO3: Illustrate the process of software design and user interface design.

CO4: Understand the basic concepts of software quality and scheduling of software projects.

CO5: Illustrate various software testing processes.

CO6: Apply project management and report preparation skills.

Unit I :

Software Process Models: Software Process, Generic Process Model - Framework Activity, Task Set and Process Patterns; Process Lifecycle, Prescriptive Process Models, Component Based Development, Aspect-Oriented Software Development, Formal Methods.Agile Process Models - Extreme Programming (XP), Adaptive Software Development, Scrum, Dynamic System Development Model, Feature Driven Development, Crystal, Web Engineering.

Unit II :

Software Requirements: Functional and Non-Functional Requirements, Eliciting Requirements, Developing Use Cases, Requirement Analysis and Modelling, Software Requirment and Specification (SRS) Document.

Unit III :

Software Design: Abstraction, Architecture, Patterns, Separation of Concerns, Modularity. Information Hiding, Functional Independence, Cohesion and Coupling, Object-Oriented Design, Data Design, Architectural Design, User Interface Design, Component Level Design.

Unit IV :

Software Quality: McCall's Quality Factors, ISO 9126 Quality Factors, Overview of -Quality Control-Quality Assurance, Risk Management, Risk Mitigation, Monitoring and Management (RMMM).Estimation and Scheduling of Software Projects: Software Sizing, LOC and FP based Estimations. Constructive Cost Model (COCOMO).

Unit V :

Software Testing: Verification and Validation, Error, Fault, Bug and Failure, Unit and Integration Tesing, White-box and Black-box Testing, Basis Path Testing, Control Structure Testing, Deriving Test Cases, Alpha and Beta Testing. Regression Testing, Performance Testing, Stress Testing,Re-engineering, Reverse Engineering.

References:

1. Pressman R.S, Software Engineering, a practitioner's approach-seventh edition, McGraw Hill
2. Ian Somerville, Software Engineering, Pearson Education Asia.
3. Jalote P, An Integrated Approach to Software Engineering, Narosa
4. Mall R, Fundamentals of Software Engineering, Prentice Hall India

CSC4E15 - Advanced Machine Learning

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand elementary mathematical concepts necessary for advanced machine learning algorithms.

CO2: Develop and model various supervised machine learning algorithms.

CO3: Understand the architecture of the Convolutional Neural Networks.

CO4: Implement recent and popular unsupervised machine learning approaches.

CO5: Understand setting up of machine learning projects and overview of reinforcement learning strategies.

CO6: Design and implement artificial neural network and deep learning architectures.

Unit I :Introduction to Elementary Linear Algebra, Probability and Machine Learning

Introduction to Linear Algebra: Matrices, Vector spaces, Orthogonality, Determinants, Overview of Eigenvalues and Eigenvectors. Fundamentals of Probability:- Axioms of probability, Conditional probability, Random variables, Probability distributions and Density functions, Joint distribution and density function, Conditional distribution, Bayes' rule, Expectation and Variance. Distributions: Bernoulli, Binomial, Multinominal, Uniform, Normal, Chi-Square, t and F. Introduction to Machine Learning:- Concept of learning the task, inductive learning and hypothesis space, different machine learning approaches, types of learning; supervised, unsupervised and reinforcement, machine learning applications

Unit II :Supervised Learning

Learning a class from example, learning multiple classes and multi-labels, model selection and generalization, linear regression and feature selection, Bayesian learning, Decision Tree learning, classification tree and regression tree, multivariate methods for learning, multivariate classification and regression

Unit III : Unsupervised Learning

clustering; mixture densities, k-means clustering, expectation maximization algorithm, mixture latent variable models, Latent Dirichlet Allocation, spectral and hierarchical clustering, Dimensionality reduction; principal component allocation, linear discriminant analysis, canonical correlation analysis.

Unit IV : Reinforcement Learning and Design and Analysis of ML Experiments

Single state case: K-Armed Bandit, Elements of reinforcement learning, model-based learning, temporal difference learning, generalization, partially observable states. Setting a machine learning platform: training, validation and testing, over-fitting and under-fitting, different types of error calculation. Cross-validation and resampling methods, Performance Analysis and accuracy measures: evaluation of machine learning algorithms, Binomial, Approximate Normal and t-tests

Unit V : Introduction to Artificial Neural Network and Deep Learning Concepts

Understanding brain, perceptron, Multi-Layer perceptron as universal approximator, the general architecture of an artificial neural network, feedforward and back-propagation, different linear and nonlinear activation functions for binary and multi-class and multi-label classification. Fundamentals of deep learning, Deep Feedforward Networks, Regularization for Deep Learning, Optimization for Training Deep Models, Introduction to Convolutional Networks, Sequence Modeling using Recurrent Nets, overview of LSTM and Bi-LSTM.

References:

1. Ethem Alpaydin, Introduction to Machine Learning- 3rd Edition, PHI.
2. Tom M. Mitchell, Machine Learning, McGraw-Hill, 1st Ed.
3. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning (Adaptive Computation and Machine Learning), MIT Press, 2016.
4. Kuntal Ganguly, Learning Generative Adversarial Networks, Packt Publishing, 2017
5. Gilbert Strang, Linear algebra and its applications, Fourth Edition, Cengage Learning, 2006.

CSC4E16 - Big Data Technologies

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Build knowledge about the importance of Big Data.

CO2: Familiarize the concept of NoSQL databases

CO3: Illustrate awareness about Hadoop Distributed File System

CO4: Understanding the concept of MapReduce

CO5: Apply Machine Learning Techniques using R.

CO6: Understand how to Access and Process Data on Distributed File System

Unit I :

Introduction to Big Data - definition and importance of Big Data - four dimensions of Big Data - volume, velocity, variety, veracity - importance of big data - structured data, unstructured data - the role of a CMS in big data management - integrating data types into a big data environment - distributed computing and Big Data. Big Data stack - layer 0,1 and 2 - Big Data management - operational databases - relational databases - non relational databases - NoSQL - key-value pair databases - document databases - columnar databases - graph databases - spatial databases.

Unit II :

Big Data analysis - basic analytics - operationalized analytics - modifying business intelligence products to handle Big Data - Big Data analytics examples - Analytics solutions - text analytics - exploring unstructured data - understanding text analytics - analysis and extraction techniques - the extracted information - text analytics tools for Big Data - custom applications for Big Data analysis - R Environment - Google Prediction API - Characteristics of a Big Data Analysis Framework.

Unit III :

NoSQL databases - types - Advantages over Relational Databases - MongoDB - introduction - MongoDB philosophy - the data model - designing the database - collections - documents - data types - the -id Field - indexes - viewing available databases and collections - opening a database

- inserting data - querying for data - retrieving documents - aggregation commands - grouping results - conditional operators - specifying an array of matches - applying criteria for search - $slice - $size - $exists - $type - $elemMatch - $not (meta-operator) - update() - save() - $inc - $set - $unset - $push - $pushAll - $addToSet - removing elements from an array - atomic operations - modifying and returning a document atomically - renaming a collection - removing data - referencing a database - implementing index-related functions - min() and max().

Unit IV :

Hadoop - history - components - HDFS - MapReduce Basics - origins of MapReduce - map function - reduce function - putting them together - Hadoop common components - application development in Hadoop - Pig and Pig Latin - Load - Transform - Dump and Store - Hive - Jaql - getting our data into Hadoop - basic copy data - Flume - Zookeeper - HBase - Oozie - Lucene - Avro.

Unit V :

Understanding MapReduce - key/value pairs - the Hadoop Java API for MapReduce - the Mapper class - the Reducer class - the Driver class - writing simple MapReduce programs - Hadoop-provided mapper and reducer implementations - Hadoop-specific data types - the Writable and WritableComparable interfaces - wrapper classes - Input/output - InputFormat and RecordReader - OutputFormat and RecordWriter. Implementing WordCount using streaming - analyzing a large dataset - summarizing the UFO data - summarizing the shape data - a relational view on data with Hive - creating a table for the UFO data - inserting the UFO data - redefining the table with the correct column separator - creating a table from an existing file - SQL views.

References:

1. Hurwitz, Alan Nugent, Fern Halper and Marcia Kaufman, Big Data for Dummies.
2. Eelco Plugge, Peter Membrey and Tim Hawkins, The Definitive Guide to MongoDB: The NOSQL Database for Cloud and Desktop Computing, 1st Edition, Apress,
3. Chris Elaton, Derk Deroos, Tom Deutsch, George Lapis and Pual Zikopoulos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, 1st Edition, Garry Turkington, Hadoop Beginner's Guide, Packt Publishing Ltd,

CSC4E17 - Mobile Communication

Course Number: 2
Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Understand the fundamental concepts of wireless and mobile networks.

CO2: Illustrate the wireless application protocols for mobile content development.

CO3: Understand TCP/IP extensions for wireless mobile networking.

CO4: Understand WAP Design and principles of operations.

CO5: Analyze various wireless mobile programming methodologies.

CO6: Understand the concept of GSM in real time applications.

Unit I :

Introduction, wireless transmission - frequencies for radio transmission - signal propagation - multiplexing - modulation - spread spectrum - cellular systems - medium access control - specialized MAC - SDMA - FDMA - TDMA - classical and slotted aloha - CSMA - collision avoidance - polling - CDMA - comparison of S/T/F/CDMA.

Unit II :

Telecommunication systems - GSM-mobile services - system architecture - radio interface - protocols - localization and calling - handover - security - new data services - satellite systems- routing- localization - handover-broadcast systems - digital audio and video broadcasting.

Unit III :

Wireless LAN-Infra red Vs radio transmission -infra structure and adhoc networks-IEEE 802.11, hyperlan- Bluetooth -IEEE 802.15

Unit IV :

Mobile network layer - mobile IP - IP packet delivery - registration - tunneling and encapsulation - optimizations - reverse tunneling - dynamic host configuration protocol-Mobile ad-hoc networks, Transport Layer-TCP-Indirect TCP-Snooping TCP-Mobile TCP- retransmission-recovery-transaction oriented TACP

Unit V :

WAP-Design and principles of operations, WAP architecture, Overview-WAP model, components-WAE, overview-WWW model-WAE model-WTA architecture, Wireless session protocol specifications-Wireless transaction protocol specification-security specification- Wireless datagram protocol-wireless control message protocol specification.

References:

1. Schiller J., Mobile Communications, Addison Wesley, 2/e, Pearson Education, 2009.
2. Gray.S.Rogers,John Edwards, An Introduction to Wireless Technology, Pearson Education
3. Singhal et.al S., The Wireless Application Protocol, Addison Wesley
4. C. Siva Ram Murthy, WDM Optical Networks: Concepts, Design, and Algorithms, Pearson Education.
5. Yi-Bang Lin and Imrich Chlamtac, Wireless and Mobile Architectures, Wiley Student Edition, 2008.
6. William Stallings, Wireless Communications and Networks, Prentice Hall, 2004
7. Vijay K.Garg, Wireless Communications and Networking, Morgan Kaufmann Publishers / Elsevier, 2009.

CSC4E18 - Internet of Things

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Familiarize fundamental concepts of IoT.

CO2: Understand roles of sensors in IoT .

CO3: Describe different protocols used for IoT design .

CO4: Familiar with data handling and analytics tools in IoT.

CO5: Understand the component parts of an IOT network and its connections to build small IOT applications.

CO6: Understand the role of IoT in various domains of Industry.

Unit I :

IOT Definitions and Functional Requirements – Motivation – Architecture - Web 3.0 View of IoT– Ubiquitous IoT Applications – Four Pillars of IoT – DNA of IoT - The Toolkit Approach for End-user Participation in the Internet of Things. Middleware for IoT: Overview – Communication middleware for IoT –IoT Information Security.

Unit II :

Protocols – IEEE 802.15.4 – BACnet Protocol – Modbus– Zigbee Architecture – Network layer – 6LowPAN - CoAP–Security. Building IoT with RASPBERRY PI vs ARDUINO:Building IOT with RASPBERRY PI- IoT Systems -

Unit III :

Logical Design using Python – IoT Physical Devices & Endpoints - IoT Device -Building blocks -Raspberry Pi -Board - Linux on Raspberry Pi - Raspberry Pi Interfaces. Programming Raspberry Pi with Python - Other IoT Platforms - Arduino.

Unit IV :

Integrating IOT: Integrated Billing Solutions in the IoT, Business Models for the IoT - Network Dynamics: Population Models – Information Cascades – Network Effects - Network Dynamics: Structural Models - Cascading Behavior in Networks - The Small World Phenomenon.

Unit V :

Case Studies and RealWorld Applications :Real world design constraints - Applications - Asset management, Industrial automation, smart grid, Commercial building automation, Smart cities - participatory sensing - Data Analytics for IoT – Software & Management Tools for IoT Cloud Storage Models & Communication APIs - Cloud for IoT - Amazon Web Services for IoT.

References:

1. Honbo Zhou, —The Internet of Things in the Cloud: A Middleware Perspective‖, CRC Press, 2012.
2. ArshdeepBahga, Vijay Madisetti, —Internet of Things – A hands-on approach‖, Universities Press, 2015
3. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), —Architecting the Internet of Things‖, Springer, 2011.
4. Jan Holler, VlasiosTsiatsis , Catherine Mulligan, Stamatis , Karnouskos, Stefan Avesand. David Boyle, "From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence", Elsevier, 2014.
5. Olivier Hersent, David Boswarthick, Omar Elloumi , —The Internet of Things – Key applications and Protocols‖, Wiley, 2012.
6. David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press - 2010.
7. Olivier Hersent, Omar Elloumi and David Boswarthick , The Internet of Things: Applications to the Smart Grid and Building Automation, Wiley .

CSC4E19 - Cyber Physical Systems

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Categorize the essential modeling formalisms of Cyber-Physical Systems (CPS)

CO2: Analyze the functional behavior of CPS based on standard modeling formalisms.

CO3: Model specific software CPS using existing synthesis tools.

CO4: Develop CPS requirements based on operating system and hardware architecture constraints.

CO5: Analyze and verify the correctness of CPS implementations against system requirements and timing constraints.

CO6: Ability to develop concepts, logics towards solving a unknown problem in research and industry

Unit I :

Introduction: Cyber-Physical System, Key Features of CPS, Application Domains of CPS, Basic principles of design and validation of CPS, Challenges in CPS. CPS Platform components: CPS HW platforms, Processors, Sensors and Actuators, CPS Network - Wireless, CAN, Automotive Ethernet, Scheduling Real Time CPS tasks.

Unit II :

Synchronous and Asynchronous Model: Reactive Components, Components Properties, Components Composing, Synchronous Designs and Circuits, Asynchronous Processes and operations, Design Primitives in Asynchronous Process, Coordination Protocols in Asynchronous Process, Leader Election, Reliable Transmission.

Unit III :

Security of Cyber-Physical Systems: Introduction to CPS Securities, Basic Techniques in CPS Securities, Cyber Security Requirements, Attack Model and Countermeasures, Advanced Techniques in CPS Securities.

Unit IV :

Advanced Automata based modeling and analysis: Basic introduction and examples ,Timed and Hybrid Automata, Definition of trajectories, zenoness, Formal Analysis: Flow pipe construction, reachability analysis, Analysis of CPS Software, Weakest Preconditions, Bounded Model checking.

Unit V :

CPS Application: Health care and Medical Cyber-Physical Systems, Smart grid and Energy CyberPhysical Systems, WSN based Cyber-Physical Systems, Smart Cities.Automotive Case study : Vehicle ABS hacking, Power Distribution Case study : Attacks on Smart grid.

References:

1. E. A. Lee and S. A. Seshia, "Introduction to Embedded Systems: A Cyber-Physical Systems Approach", 2011.
2. R. Alur, "Principles of Cyber-Physical Systems," MIT Press, 2015
3. T. D. Lewis "Network Science: Theory and Applications", Wiley, 2009.
4. P. Tabuada, "Verification and control of hybrid systems: a symbolic approach", Springer-Verlag 2009.
5. C. Cassandras, S. Lafortune, "Introduction to Discrete Event Systems", Springer 2007.
6. Constance Heitmeyer and Dino Mandrioli, "Formal methods for real-time computing", Wiley publisher, 1996.
7. Raj Rajkumar, Dionisio de Niz and Mark Klein, "Cyber-Physical Systems", Addison-Wesley, 2017 .

CSC4E20 - Blockchain Technology

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: To Understand and apply the fundamentals of Cryptography in Crypto currency

CO2: To gain knowledge about various operations associated with the life cycle of Blockchain and Crypto currency.

CO3: Contentedly discuss and describe the history, types and applications of Blockchain

CO4: To acquire knowledge about cryptography and consensus algorithms.

CO5: Implement an ICO on Ethereum

CO6: Design Blockchain based application with Swarm and IPFS

Unit I :

Introduction to Blockchain :Distributed DBMS – Limitations of Distributed DBMS, Introduction to Block chain – History, Definition, Distributed Ledger, Blockchain Categories – Public, Private, Consortium, Blockchain Network and Nodes, Peer-to-Peer Network, Mining Mechanism, Generic elements of Blockchain, Features of Blockchain, and Types of Blockchain.

Unit II :

Blockchain Architecture:Operation of Bitcoin Blockchain, Blockchain Architecture – Block, Hash, Distributer P2P, Structure of Blockchain-Consensus mechanism: Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT), Proof of Authority (PoA) and Proof of Elapsed Time (PoET) .

Unit III :

Blockchain-based Future System:Project presentation- Futures smart contract: Blockchain oracles- Web3j: Setting up the Web3J- Installing web3j- Wallet creation, Java client: The wrapper generator- Initializing web3j- Setting up Ethereum accounts- Deploying the contract.

Unit IV :

Blockchains in Business and Creating ICO :Public versus private and permissioned versus permission less blockchains- Privacy and anonymity in Ethereum- Why are privacy and anonymity important? - The Ethereum Enterprise Alliance- Blockchain As-a-Service- Initial Coin Offering (ICO): Project setup for ICO implementation- Token contracts- Token sale contracts- Contract security and testing the code.

Unit V :

Distributed Storage IPFS and SWARM :Ethereum Virtual Machine-Swarm and IPFS: Installing IPFS, Hosting our frontend: Serving your frontend using IPFS, Serving your frontend using Swarm, IPFS file uploader project: Project setup the web page.

References:

1. Imran Bashir, "Mastering Blockchain: Distributed Ledger Technology, decentralization, and smart contracts explained", 2nd Edition, Packt Publishing Ltd, March 2018.
2. Bellaj Badr, Richard Horrocks, Xun (Brian) Wu, "Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger", Packt Publishing Limited, 2018.
3. Andreas M. Antonopoulos , "Mastering Bitcoin: Unlocking Digital Cryptocurrencies", O'Reilly Media Inc, 2015.
4. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", Princeton University Press, 2016.
5. Franco, P. (2014). Understanding Bitcoin: Cryptography, engineering and economics. John Wiley and Sons.
6. Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press.
7. William Stallings, Cryptography and Network Security, Pearson 2004.
8. Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies.OReilly Media, Inc."

CSC4E21 - Quantum Computation

Contact Hours per Week: 4 (3 Lecture + 1 Tutorial)
Number of Credits: 4
Course Outcomes:

CO1: Demonstrate proficiency with the mathematics behind quantum algorithms, such as unitary operators and quantum gates.

CO2: Understand the major quantum computing algorithms, such as Grover and Shor.

CO3: Understand Quantum computational complexity and error correction mechanisms.

CO4: Learn the framework of quantum computation, and how that may be useful for future quantum technologies.

CO5: Understand Quantum Computational Complexity and Error Correction.

CO6: Model basic Quantum Circuits.

Unit I :

Introduction:Overview of traditional computing – Church-Turing thesis – circuit model of computation– reversible computation – quantum physics – quantum physics and computation – Dirac notation and Hilbert Spaces – dual vectors operators – the spectral theorem –functions of operators – tensor products – Schmidt decomposition theorem.

Unit II :

QUBITS and Quantum Model of Computation :State of a quantum system – time evolution of a closed system – composite systems –measurement – mixed states and general quantum operations – quantum circuit model –quantum gates – universal sets of quantum gates – unitary transformations – quantum circuits.

Unit III :

Quantum Algorithms:Superdense coding – quantum teleportation – applications of teleportation – probabilistic versus quantum algorithms – phase kick-back – the Deutsch algorithm – the Deutsch-Jozsa algorithm – Simon's algorithm – Quantum phase estimation and quantum Fourier Transform – eigenvalue estimation.

Unit IV :

Order-finding problem – eigenvalue estimation approach to order finding – Shor's algorithm for order finding – finding discrete logarithms – hidden subgroups – Grover's quantum search algorithm – amplitude amplification – quantum amplitude estimation –quantum counting – searching without knowing the success probability.

Unit V :

Quantum Computational Complexity and Error Correction:Computational complexity – black-box model – lower bounds for searching – general black-box lower bounds – polynomial method – block sensitivity – adversary methods –classical error correction – classical three-bit code – fault tolerance – quantum error correction – three- and nine-qubit quantum codes – fault-tolerant quantum computation.

References:

1. P. Kaye, R. Laflamme, and M. Mosca, "An introduction to Quantum Computing",Oxford University Press, 1999.
2. V. Sahni, "Quantum Computing", Tata McGraw-Hill Publishing Company, 2007.
3. Michael A. Nielsen, Issac L. Chuang, "Quantum Computation and Quantum Information", Tenth Edition, Cambridge University Press, 2010.
4. Scott Aaronson, "Quantum Computing Since Democritus", Cambridge University Press, 2013.
5. N. David Mermin, "Quantum Computer Science: An Introduction", Cambridge University Press, 2007.

CSC4C17: Project Work & Dissertation

Course Number: 3
Number of Credits: 8
Course Outcomes:

CO1: Understand programming language concepts and software engineering principles to develop a medium sized software projects for industry or propose any new model for the selected field of research.

CO2: Understand the project requirements, reflect on their learning and take appropriate actions to implement it.

CO3: Communicate effectively and to present ideas clearly and coherently to specific audiences in both the written and oral forms.

CO4: Understand the importance of iteration, evaluation and prototyping in design of a software system.

CO5: Estimate, plan, calculate, and adjust project variables.

CO6: Innovate, experiment and analyze research findings and practice the process of scientific publishing.

Course Outline
Major project work is to be done individually by each student, under the guidance of a faculty member of the Department. Project Supervisor has to constantly monitor the works done by the student, imparting him/her the necessary inputs for the successful completion of the project work. Students can either take up real-life application-oriented project work or research and development projects. The student can formulate a project problem with the help of her/his guide and submit the project proposal of the same. Approval of the project proposal is mandatory. If approved, the student can commence working on it, and complete it.

Guidelines for Submission of Report
The distinguishing mark of a dissertation is an original contribution to knowledge. The dissertation is a formal document whose sole purpose is to prove that you have made an original contribution to knowledge. Failure to prove that you have made such a contribution generally leads to failure. It is a test of the student's ability to undertake and complete a sustained piece of independent research and analysis / application development, and to write up the work in a coherent form according to the rules and conventions of the academic

community. The role of the supervisor too is very crucial in this context.
A satisfactory dissertation should not only be adequate in its methodology, in its analysis and in its argument, and adequately demonstrate its author's familiarity with the relevant literature; it should also be written in correct, coherent language, in an appropriate style, correctly following the conventions of citation. It should, moreover, have a logical and visible structure and development that should at all times assist the reader understand the arguments being presented. The layout and physical appearance of the dissertation should also conform to university standards.The dissertation is to be prepared in TEX format (either Latex or a suitable Windows TEX variant).
Students are also encouraged to present their work in Seminar/ conference/ workshop/journal with the assistance and guidance of the Project supervisor. This should pave as a good start for the student in the art of publishing/ presenting his/her work to the outside world. Due weightage is accommodated for publications out of the project work in the final evaluation.

****

General Pattern of Question Paper

Core and Elective courses in M.Sc. Computer Science Programme

Under CCSS (with effect from 2022 Admission onwards)

Code: Reg. No:

Name:

1st/2nd/3rd/4th Semester M.Sc. Computer Science Degree Examination – 2022

(CCSS – M.Sc. Programme)

Course Code : (eg: CSC1C02) Course : (Eg: Advanced Data Structures and Algorithms)

Time: 3 Hours **Total Marks: 50**

Answer five full questions; Each Question carries 10 marks.

Question Numbers:1 to 8 Total Marks = 5 x 10 = 50 Marks

NOTE: Minimum one question from each of the five modules. Remaining three questions can be from any module. There should not be more than two questions from the same module.