FAROOK COLLEGE (AUTONOMOUS)

ORA ET LABORA

CHOICE BASED CREDIT SEMESTER SYSTEM-PG
(FCCBCSS PG – 2019)

M.Sc. Computer Science

REGULATIONS and SYLLABUS

FOR THE POST GRADUATE PROGRAMMES OF FAROOK COLLEGE (AUTONOMOUS)

CERTIFICATE

I hereby certify that the documents attached are the bona fide copies of the syllabus of M.Sc. Computer Science Programme to be effective from the academic year 2019-20 onwards.

Date:

Place: PRINCIPAL

PROGRAMME STRUCTURE FOR MSc COMPUTER SCIENCE
(EFFECTIVE FROM THE ACADEMIC YEAR 2019- 20)

LEGEND

Item	Description
C	Credits
E	External Component (%)
I	Internal Component (%)
L	Lecture Hours
P	Practical Hours
T	Total

SEMESTER I

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
1.1	MCS1C01	Discrete Mathematical Structures	4	1	4	5	4	0	4
1.2	MCS1C02	Advanced Data Structures	4	1	4	5	4	1	5
1.3	MCS1C03	Theory of Computation 0	4	1	4	5	4	0	4
1.4	MCS1C04	The Art of Programming Methodology	4	1	4	5	4	0	4
1.5	MCS1C05	Computer Organization Architecture	4	1	4	5	4	0	4
1.6	MCS1L01	Practical I	2	1	4	5	0	4	4
1.7	MCS1A01	Introduction to Research (Ability Enhancement Audit Course)	4	5	0	5	0	0	0
		Total Credits (Excluding Audit Course): 22					**20**	**5**	**25**

SEMESTER II

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
2.1	MCS2C01	Design and Analysis of Algorithms	4	1	4	5	4	0	4
2.2	MCS2C02	Operating System Concepts	4	1	4	5	4	1	5
2.3	MCS2C03	Computer Networks	4	1	4	5	4	0	4
2.4	MCS2C04	Artificial Intelligence	4	1	4	5	4	0	4
2.5	MCS2C05	Principles of Software Engineering	4	1	4	5	4	0	4
2.6	MCS2L02	Practical II	2	1	4	5	0	4	4
2.7	MCS2A02	Term Paper (Professional Competency Audit Course)	4	5	0	5	0	0	0
		Total Credits (Excluding Audit Course): 22					**20**	**5**	**25**

SEMESTER III

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
3.1	MCS3C01	Advanced Database Management System	4	1	4	5	4	1	4
3.2	MCS3C02	Principles of Compilers	4	1	4	5	4	0	5
3.3	MCS3C03	Object Oriented Programming Concepts	4	1	4	5	4	0	4
3.4	MCS3E01	Elective I	4	1	4	5	4	0	4
3.5	MCS3E02	Elective 2	4	1	4	5	4	0	4
3.6	MCS3L03	Practical III	2	1	4	5	0	4	4
		Total Credits (Excluding Audit Course): 22					20	5	25

List of Elective Courses for MCS3E01	
Course Code	**Course Name**
MCS3E01a	Computer Graphics
MCS3E01b	Introduction to Soft Computing
MCS3E01c	Web Technology
MCS3E01d	Bioinformatics
MCS3E01e	Computer Optimization Techniques
MCS3E01f	Numerical and Statistical Methods

List of Elective Courses for CSS3E02	
Course Code	**Course Name**
MCS3E02a	Pattern Recognition
MCS3E02b	Wireless and Mobile Networks
MCS3E02c	Cryptography and Network Security
MCS3E02d	Advanced Web Technology
MCS3E02e	Virtualisation and Cloud Computing
MCS3E02f	Data Warehousing and Data Mining

SEMESTER IV

No	Course Code Course Name		C	Weightage			Hrs/Week		
				I	E	T	L	P	T
4.1	MCS4E03	Elective 3	3	1	4	5	5	0	5
4.2	MCS4E04	Elective 4	3	1	4	5	5	0	5
4.3	MCS4P01	Project Requirements Analysis & Design Related Discussion	8	1	4	5	3	1	4
		Project Coding, Testing & Implementation Related Discussion					2	2	4
		Project Evaluation & Assessment					2	0	2
		Project Lab Work					0	5	5
		Total Credits (Excluding Audit Course): 14					**17**	**8**	**25**

List of Elective Courses for MCS4E03	
Course Code	**Course Name**
MCS4E03a	Data Compression
MCS4E03b	Pervasive Computing
MCS4E03c	System Security
MCS4E03d	Molecular Simulation and Modelling
MCS4E03e	Fundamentals of Big Data
MCS4E03f	Web Engineering

List of Elective Courses for MCS4E04	
Course Code	**Course Name**
MCS4E04a	Digital Image Processing
MCS4E04b	Introduction to block chain technology
MCS4E04c	Software Development for Portable Devices
MCS4E04d	Storage Area Networks
MCS4E04e	Semantic Web
MCS4E04f	Advanced Java Programming

MCS1C01 Discrete Mathematical Structures

Course Number: 1.1
Contact Hours/Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To introduce discrete mathematics concepts necessary to understand basic foundation of Computer Science.

Course Outcome

Students will be able to understand and apply fundamental mathematical principles required for computer science

Course Outline

Unit I

Sets and Mathematical Logic: Set Theory - Types of sets, Set operations, Principles of Inclusion and Exclusion. Mathematical Logic - Propositional Calculus - Statement, Connectives, Conditional and Biconditional, Equivalence of Formula, Well Formed Formula, Tautologies, Duality Law, Functionally Complete Sets of Connectives, Normal Forms, Theory of Inference for the Statement Calculus, Predicate Calculus - Statement Functions, Variables and Quantifiers, Free and Bound Variables, Theory of Inference for the Predicate Calculus.

Unit II

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 III

Lattices and Boolean Algebra - Lattices and Algebraic Systems, Principles of Duality, Basic Properties of Algebraic Systems Defined by Lattices, Distributive Lattices and Complemented Lattices. Boolean Lattices and Boolean Algebras. Boolean Functions and Boolean Expressions.

Unit IV

Group Theory – Definition and Elementary Properties - Permutation Groups, Cyclic Groups – Subgroups - Cosets and Lagrange’s Theorem, Semigroup and Monoid. Homeomorphism and Isomorphism. Rings, Integral Domains and Fields.

Unit V

Graph Theory – Introduction, Directed Graph, Undirected Graph, Connected and Disconnected Graphs, Bipartite Graph, Complete Bipartite Graph, Isomorphic Graphs, Subgraph. Paths and Circuits. Shortest Paths in Weighted Graphs - Dijkstra's Algorithm. Eulerian Paths and Circuits, Hamiltonian Paths and Circuits. Trees - Spanning Trees and Cut-Sets, Minimum Spanning Trees - Kruskal's Algorithm, Prim's Algorithm.

Reference

1. C Liu and D. Mohapatra, *Elements of Discrete Mathematics - A Computer Oriented Approach*, TMH, ISBN: 1259006395.
2. Alan Doerr and Kenneth Levassur, *Applied Discrete Structure for Computer Science*, Galgotia Publications Pvt. Ltd, ISBN: 9780574217554.
3. J. K. Sharma, *Discrete Mathematics*, Macmillan Publishers India Limited, ISBN: 1403924759.
4. J. P. Tremblay and R. Manohar, *Discrete Mathematical Structures with Application to Computer Science,* McGraw-Hill Companies, ASIN: B001FPXR5Y.

MCS1C02 | Advanced Data Structures

Course Number: **1.2**
Contact Hours per Week: **5 (4 Lecture + 1 Practical)**

Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20 % (Internal) + 80 % (External)**

Objectives

- To introduce basic and advanced data structures dealing with algorithm development and problem solving.

Course Outcome

Student will be able to understand and implement basic and advanced data structures for problem solving.

Course Outline

Unit I

Data structure – definition - types & operations, characteristics of data structures - Abstract Data Type (ADT) – algorithms – concepts – definition - objectives of algorithms - quality of an algorithm - space complexity and time complexity of an algorithm.

Unit II

Counting Techniques: Basic counting techniques - permutations and combinations, asymptotic behavior of functions. Linear data structures - Arrays – records – representation - data structure operations - traversing, inserting and deleting - sorting and searching - sorting algorithms - linear search & binary search – complexity. Linked lists – operations and implementations, Stack - operations and its implementations (both array and linked list) – Applications - parsing arithmetic expressions, conversion and evaluating expressions. Recursion - characteristics of recursion, types of recursion - applications of recursion in algorithms - comparison of recursive and non-recursive algorithms. Queue - operations and its implementations (both array and linked list) – circular queue – dequeue - priority queues, recursive lists, heterogeneous lists, deterministic skip lists, doubly linked lists and circular lists - sparse matrix- representation.

Unit III

Non- linear Data Structures - trees – terminology - tree traversals algorithms - Binary trees - threaded binary trees – binary search trees - traversals and operations on BST – heap Tree - balanced trees - M-way trees – B and B+ trees, Red Black Tree, Digital Search Tree, Tries, Treaps, Huffman algorithm for extended binary tree – operations and their implementation. Graphs - representation of graphs - operations - traversals 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 – rehashing – extendable hashing - separate chaining - hashing efficiency – heaps - overview of heaps - implementation and operations.

Unit V

Heap structures - Min- Max heaps - Deaps - leftist heaps - binomial heaps - Fibonacci heaps -binary heaps - skew heaps - pairing heaps – applications - amortized analysis-an unrelated puzzle - Binomial queues - skew heaps - Fibonacci heaps - Splay trees.

References:

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, ISBN: 978-0201000238.
2. Horowitz E and Sahni S, *Fundamentals of Data Structures*, Computer Science Press, ISBN: 9780716780427.

3. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, *Fundamentals of Data Structures in C*, Silicon Press, ISBN: 0929306406.

4. Richard F. Gilberg and Behrouz A. Forouzan, *Data Structures: A Pseudocode Approach With C*, Thomson Brooks/Cole Publications, Course Technology, ISBN: 9780534390808.

5. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, *Data Structure using C*, Prentice- Hall, ISBN: 9780131997462.

6. Robert Kruse, Tondo C L and Bruce Leung, *Data Structures & Program Design in C*, Pearson India, 2nd Edition, ISBN: 9788177584233.

7. U. A. Deshpande and O. G. Kakde, *Data Structures & Algorithms*, ISTE Learning Materials Centre, New Delhi, ISBN: 9788188057054.

8. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, *Introduction to Algorithms*, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN: 978-0262033848.

9. Seymour Lipschutz, *Data Structures With C*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070701989.

10. Jean-Paul Tremblay, Paul G. Sorenson, P. G. Sorenson, *Introduction to Data Structures with Applications*, 2nd Edition, Mcgraw-Hill College, ISBN: 0070651574.

MCS1C03 | Theory of Computation

Course Number: **1.3**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

□ To learn different types of machines and corresponding grammars.
□ To learn the undecidability of problems
□ To familiarize the complexity classes P and NP.

Course Outcome

Students will be able to

- Construct appropriate grammars/machines for a given language.
- Classify languages according to Chomsky Hierarchy
- Classify the problems based on complexity classes P and NP.

Course Outline

Unit I (12 hours)

Preliminaries - Introduction to formal proof and inductive proofs - The central concepts of Automata Theory - Alphabets, Strings, Languages – Introduction to automata and grammar - Deterministic Finite Automata, Non- deterministic Finite Automata – Equivalence of Deterministic and Nondeterministic Finite Automata - Finite Automata with Epsilon Transitions - Equivalence of NFA with and without epsilon moves.

Unit II (12 hours)

Regular Expressions, Finite Automata and Regular Expressions, Properties of Regular Languages - Pumping lemma and proof for existence of non regular languages, Closure properties, homomorphism, substitution - Myhill Nerode Theorem and DFA state minimization – Regular Grammar.

Unit III (12 hours)

Context Free Languages - Equivalence of CFG and PDA – Normal forms (CNF and GNF) – Closure properties of CFL's – DCFL's and their properties – Decision procedures – CYK algorithm – Pumping lemma and proof for existence of non context- free languages – Context sensitive languages: Equivalence of LBA and Context Sensitive Grammar (CSG).

Unit IV (12 hours)

Turing machines - TM computations – Equivalence of standard TM with multi tape and non deterministic TM's – Turing acceptable, Turing decidable and Turing enumerable language classes - Equivalence of type 0 grammars with TM's – Church thesis – Chomsky hierarchy - Closure properties of recursive and recursively enumerable Languages.

Unit V (12 hours)

Computability and Decidability – halting problem – reductions – post correspondence problem. Computational complexity - Time and space bounded simulations – Classes P and NP – NP completeness.

References

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, *Introduction to Automata Theory, Languages of Computation*, 3rd Edition, Prentice Hall, ISBN: 0321455363.
2. Linz P, An *Introduction to Formal Languages and Automata*, Narosa Publishing House Pvt. Ltd., New Delhi, ISBN: 9788173197819.
3. Michael Sipser, *Introduction to Theory of Computation*, Cengage Learning India Private Limited, Indian Edition, ISBN: 8131505138.
4. H.R. Lewis and C.H. Papadimitriou, *Elements of Theory of Computation*, 2nd Edition, Prentice Hall, ISBN: 0132624788.
5. J. E. Savage, *Models of Computation, Exploring the Power of Computing*, Addison Wesley, 1998, Available at http://cs.brown.edu/~jes/book/.
6. Martin J.C, Introduction *to Languages and Theory of Computation*, Tata McGraw Hill, 3rd Edition, ISBN: 9780070660489.

MCS1C04| The Art of Programming Methodology

Course Number: 1.4
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure: None.
Course Evaluation: 20 % (Internal) + 80 % (External)

Objectives

To learn the art of designing algorithms and flowcharts.
To introduce the concept of algorithmic approach for solving real-life problems.
To develop competencies for the design and coding of computer programs.
To learn designing programs with advanced features of C.

Course Outcomes

Student will be able to

- Understand problem solving using flowcharts and algorithms
- Develop flowcharts / algorithms for problems using sequence, selection and iterative constructs and implement them using C programming

Unit I

Part A: Problem Solving – Flow Chart for Structured Programming – Program Charts – System Charts – Variables, data names, programming statements – Conventions for Drawing Flow Charts - Flow Chart Symbols - **Part B**: Algorithm Design – Problem Solving Aspect – Top Down Design – Formal

Conventions – Structure of Algorithms – Case Study: Flow charts for problems with only sequence constructs: Adding two numbers, swapping numbers etc. **Part C**: 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 - Evolution and Features of C - C Program Structure - Elements of C Language and Program constructs - character set, tokens, keywords, identifier - Data types, constants, symbolic constants, variables, declaration, data input and output, assignment statements. Storage Classes. Operators in C - arithmetic, relational, logical, assignment, auto increment, auto decrement, conditional, comma operators. Precedence of operators - expressions – evaluation of expressions, type conversion in expressions – precedence and associativity. Case Study: Coding of flowcharts and algorithms developed in Unit I.

Unit III

Decision making – IF, IF ELSE, IF ELSE IF LADDER, SWITCH, BREAK, CONTINUE, GOTO. Case study: Developing flowcharts, algorithms and coding for problems that requires selection constructs - Converting a solution using one selection construct to another one (eg: replace IF ELSE with IF / SWITCH).

Unit IV

Looping - WHILE, DO-WHILE, and FOR - Nesting of loops - skipping & breaking loops. Arrays - 1D & 2D arrays - strings – processing of strings - string manipulation functions. Case study: Developing flowcharts, algorithms and coding for problems that requires selection and/or iteration constructs - Converting a solution using one iterative construct to another one (eg: replace FOR with WHILE) - Study of computing current term from the previous term for iterative solutions.

Unit V

The Concept of modularization - scope and lifetime of variables - defining function - types of functions – User defined functions - function prototype and definition – arguments - passing parameters - call by reference - call by value – returning - nesting of functions. Recursion - recursion vs. iteration, tail recursive functions and non tail recursive functions. Case study: Developing flowcharts, algorithms and coding for problems using the non recursive functions and recursive functions.

References

1. J.B Dixit, Computer Fundamentals and Programming in C, Firewall Media, ISBN: 8170088828. (Unit I Part A, C).
2. Martin M. Lipschutz and Seymour Lipschutz, Schaum's Outline of Theory and Problems of Data Processing, ISBN: 9780070379831 (Unit I Part A).
3. Anil Bikas Chaudhuri, The Art Of Programming Through Flowcharts & Algorithms, Laxmi Publications, New Delhi (Unit I Part A).
4. Jean Paul Trembley and Pual G Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill (Unit I Part B).
5. R G Dromey, How to Solve by Computer, Pearson Education, 5th Edition, ISBN: 0134340019 (Unit I Part B).
6. Dennie Van Tassel, Program Style, Design, Efficiency, Debugging, and Testing, PHI, ISBN: 0137299478 (Unit I Part C).
7. E Balagruswamy, Programming in ANSI C, TMH, 5th Edition, ISBN: 0070681821.
8. Kamthane, Programming in C, 2nd Edition, Pearson India, ISBN: 8131760316.
9. Brian W. Kernighan and Dennis M. Ritchie, C Programming Language, PHI, ISBN: 0131103628.
10. Kanetkar, Let Us C, BPB Publications, 8th Edition, ISBN: 1934015253.

MCS1C05 | Computer Organization & Architecture

Course Number: 1.5
Contact Hours per Week: 4
Number of Credits: 4

Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

To familiarize with the digital fundamentals, computer organization, computer architecture and assembly language programming.

Course Outcome

Students will be able to

- apply digital electronics fundamentals.
- describe basic organization of computer and the architecture of 8086 microprocessor.

Course Outline

Unit I (12 Hours)
Logic gates - flip-flops –SR,D,JK,T- design of combinational and sequential circuits - examples of digital circuits – adders, multiplexers, decoders, counters- Asynchronous (Ripple) Counter, Synchronous Counter, shift registers- Parallel, serial and universal shift registers.

Unit II (12 Hours)
Basic computer organization – machine instructions – classification, function, addresses, size, addressing modes – instruction cycle - instruction sequencing. Fundamental concepts – registers, register transfers, memory read and write, execution of a complete instruction, branch instruction, bus organizations, a complete processor - Control unit - hardwired control, micro programmed control, micro instructions-types.

Unit III(12 Hours)
Arithmetic & Logic Unit - addition of positive numbers – fast adders – signed addition and subtraction - addition/subtraction logic unit – multiplication of positive numbers
– array multiplier, sequential multiplier - signed number multiplication - multiplication using Booth's algorithm - division-restoring and non-restoring algorithms, floating point numbers and operations.

Unit IV(12 Hours)
Main Memory - memory hierarchy – main memory – RAM, ROM - cache memory – virtual memory - 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(12 Hours)
Architecture - General 8-bit microprocessor and its architecture - 8085 - Functional block diagram – architecture functions of different sections - architecture of 8086 CPU. Instruction Sets - Instruction format - addressing modes - instruction set of 8085 CPU - Instruction cycle - timing diagrams - different machine cycles - fetch and execute operations.

References

1. V Carl Hamacher, Zvonko Vranesic and Safwat Zaky, *Computer Organization*, Mc-Graw Hill International Edition, 5th Edition, ISBN: 9780071122184.
2. Morris Mano, *Digital Logic and Computer Design*, Prentice Hall of India, ISBN: 0876924178.
3. M Morris Mano, *Computer System Architecture*, Prentice Hall, 3rd Edition. ISBN: 0131755633.
4. William Stallings, *Computer Organization and Architecture*, 9th Edition, Prentice Hall, ISBN: 013293633X.

5. Andrew S Tanenbaum, *Structured Computer Organization*, Prentice Hall, 6th Edition, ISBN: 0132916525.
6. Floyd Thomas L, *Digital Fundamentals*, Pearson Education, 10th Edition, Prentice Hall, ISBN: 0132359235.
7. Albert Paul Malvino, Donald P Leach, *Digital Principles and Applications*, McGraw Hill, 4th Edition, ISBN: 0070398836.
8. Thomas C Bartee, *Digital Computer Fundamentals*, McGraw Hill, 6th Edition, ASIN: B004H0SL5K.
9. Ramesh. S. Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, 6th Edition, Wiley Eastern Ltd, New Delhi, ISBN: 9788187972884.
10. Mohamed Rafiquzzaman, *Introduction to Microprocessors and Microcomputer Based System Design*, 2nd Edition, CRC Press, ISBN: 9780849344756.
11. Muhammad Ali Mazidi, Janice Mazidi, Rolin Mckinlay, Janice M. Mazidi, Janice Gillispie Mazidi and Rolin D., *The 8051 Microcontroller and Embedded Systems*, Pearson Education Asia, 5th Indian Reprint, ISBN: 013119402X.

MCS1L01 | Practical I

Course Number: 1.6

Contact Hours per Week: 4

Number of Credits: 2

Number of Contact Hours: 60 Hrs

Prerequisite/Exposure: None

Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To practically implement the theory portions covered in *The Art of Programming Methodology (MCS1C04)* and *Advanced Data Structures (MCS1C02).*

Course Outcome

Student will be able to develop programs in C and implement different data structures in C.

Course Outline

Unit I: C Programming

1. Simple C Programs like area of a circle, checking whether a given number is odd or even.
2. Implementation of programs using Loops (pyramid printing, factorial computation, number reversing, checking for Armstrong numbers, finding first N or Nth Prime numbers etc).
3. Use of 1D and 2D Arrays (searching, sorting and vector operations, matrix addition, matrix multiplication).
4. String Manipulations.
5. Structures and Unions (like addition of two complex numbers, student record creation and manipulation etc).
6. Writing functions.
7. Implementation of recursion (recursive function to compute a factorial, reverse string etc).
8. Command line arguments.
9. Pointers - simple programs to learn concept of pointers, array operation using pointers etc.
10. File operations – file and structures.

Unit 2: Data Structures and Algorithms

1. Implementation of stacks using arrays.
2. Implementation of queues, circular queue using arrays.
3. Implementation of sequential search and binary search techniques.
4. Implementation of linked lists and operations (add, insert, delete, search) on linked lists.
5. Implementation of stacks using linked list.
6. Implementation of queues using linked list.
7. Implementation of doubly linked list.
8. Implementation of circular linked list.
9. Implementation of binary tree and traversals.
10. Implementation of Binary search trees and perform the operations on BST.

11. Implementation of various sorting algorithms.
12. Conversion of an infix expression to the postfix form using stacks.
13. Evaluation of a postfix expression.
14. Implementation of graphs and graph traversals.
15. Implementation of heap tree and operations.

MCS1A01 – INTRODUCTION TO RESEARCH (ABILITY ENHANCEMENT AUDIT COURSE)

Objectives: Large numbers of students are actively considering and taking up research and associated higher studies. An introductory course on research aims to introduce students to the important aspects of research. The intent of such a course is to make students aware of the details associated with formal research. By going through this introductory course on research, students are likely to be able to take up research activities in a more systematic and formal manner right from the beginning. The specific objectives of the course include:

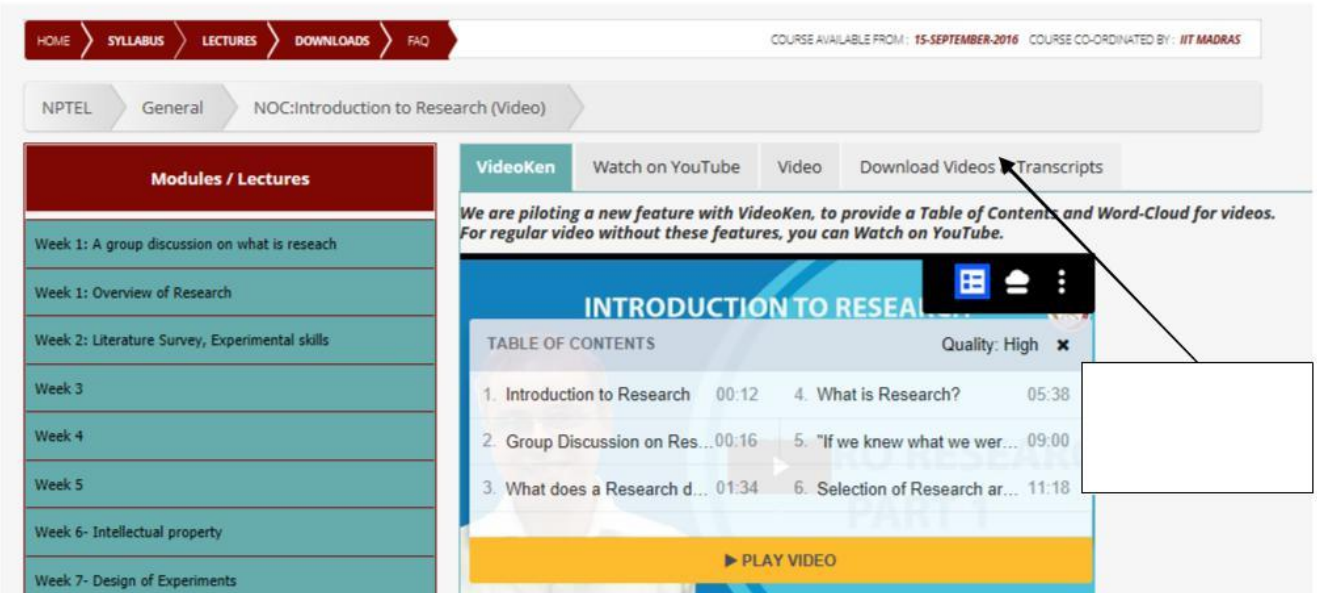
- Understand research terminology
- Be aware of the ethical principles of research
- Identify the components of a literature review process
- Critically analyze published research
- To introduce research methods in the field of computer Science

Course Evaluation & Course Credit

The Ability Enhancement Audit Course has 4 credits which will not be counted for evaluating the overall SGPA & CGPA. The College/Department shall conduct examination of 2 Hrs duration with a minimum of 20 weightage before the conclusion of first semester classes 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 this Audit Course.

Course Delivery Mode

This course is an Ability Enhancement Audit Course. The course content is not delivered in the classrooms. Instead, the students have enrol themselves for the online course offered at NPTEL. The online course is available at https://nptel.ac.in/courses/121106007/. Students can either view the video module online or can download the video lessons and transcripts to view or read them offline.



COURSE AVAILABLE FROM : 15-SEPTEMBER-2016 COURSE CO-ORDINATED BY : IIT MADRAS

Week 1: A group discussion on what is reseach
Week 1: Overview of Research
Week 2: Literature Survey, Experimental skills
Week 3
Week 4
Week 5
Week 6- Intellectual property
Week 7- Design of Experiments

We are piloting a new feature with VideoKen, to provide a Table of Contents and Word-Cloud for videos. For regular video without these features, you can Watch on YouTube.

TABLE OF CONTENTS Quality: High

1. Introduction to Research 00:12
2. Group Discussion on Res... 00:16
3. What does a Research d... 01:34
4. What is Research? 05:38
5. "If we knew what we wer... 09:00
6. Selection of Research ar... 11:18

Course Outline

The students are encouraged to cover the following modules of the course *Introduction to Research* from NPTEL:

- Week1: Overview of Research
- Week2: Overview of Literature Survey: Literature Survey using Web of Science, Literature Survey using Scopus, Writing Up, Tutorial on using BibTeX with LaTeX to add references to a document, Tutorial on using Microsoft Word with Bibliographic Sources, Tutorial on using Microsoft Word with endnote entries
- Week3: Data Analysis
- Week4: How to make Technical presentation – Technical Writing
- Week 6: Intellectual property
- Week8: Research in Computer Science & Engineering

References:

Video Lessons and Transcripts available (including in the regional language) at
https://nptel.ac.in/courses/nptel_download.php?subjectid=121106007

SEMESTER II

MCS2C01 | Design and Analysis of Algorithms

Course Number: **2.1**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives of the Course

- Understand techniques for effective problem solving in computing.
- Illustrate the use of different paradigms of problem solving.
- Understand the analysis of the algorithm.

Prerequisites

- Data Structures, Discrete Mathematics.

Course Outcome

- Understand the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe and synthesize different paradigms and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm.

Module I (12 Hours)

Algorithm Design: Introduction, Steps in developing algorithm, Methods of specifying an algorithm, Decisions prior to designing: based on the capabilities of the device, based on the nature of solutions, based on the most suitable data structures. Model of Computation: RAM model and PRAM model. Important Problem Types (Introductory concepts): Sorting, Searching, String processing, Graph problems, combinatorial problems, Geometric problems and Numerical problems.

Module II (12 Hours)

Algorithm Analysis: Importance of algorithm analysis, Time and Space Complexity. Growth of Functions: Asymptotic notations, Cost estimation based on key operations- Big Oh, Big Omega, Little Oh, Little Omega and Theta notations, Big Oh Ratio Theorem, Big Theta Ratio Theorem, Big Omega Ratio Theorem. Analyzing Algorithm Control Structures, Solving Recurrences: Iteration Method, Substitution Method, The Recursion Tree Method, Master‘s Theorem, Problem solving using Master‘s Theorem Case 1, Case 2 and Case 3. Analysis of Strasser‘s algorithm for matrix multiplication, Analysis of Merge sort.

Module III (12 Hours)

Basic Technique for Design of Efficient Algorithm: Brute Force approach (String matching), Divide-and-Conquer approach (Merge sort), Branch-and-Bound technique (Knapsack problem). Greedy approach (Kruskal‘s algorithm and Prim‘s Algorithm), Dynamic Programming (Longest Common Subsequence),Backtracking (Sum of subsets problem).

Module IV (10 Hours)

Complexity - Complexity Classes: P, NP, NP Hard and NP Complete problems. NP Completeness reductions for Travelling Salesman Problem and Hamiltonian Cycle. P versus NP problem.

Module V (14 Hours)

Analyzing Parallel Algorithms: Time Complexity, Cost, Number of Processors, Space Complexity, Speed up, Efficiency, Scalability, Amdahl‘s Law. Parallel merging and sorting, Euler tour technique, Parallel prefix computation, Deterministic symmetry breaking.

References:

1. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, Introduction to Algorithms, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN: 9780262033848 (Unit I, II, III and IV).

2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, The Design and Analysis of Computer Algorithms, 1st Edition. Addison Wesley, ISBN: 0534915728 (Unit I, II, III and IV).

3. Pallaw, V K, Design and Analysis of Algorithms, Asian Books Private Ltd, 2012, ISBN: 8184121687(UnitI, II, III and IV).

4. Sanjay Razdan, Fundamentals of Parallel Computing, Narosa Publishing House, 2014, ISBN: 9788184873481 (Unit V).

5. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013, ISBN: 9788131803349 (Unit I, II, III and IV).

6. Upadhyay N, Design and Analysis of Algorithms, SK Kataria & Sons, 2008 (Unit I, II, III and IV).

7. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley, ISBN: 9780201003277 (Unit I, II, III and IV).
8. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India, ISBN: 0133350681 (Unit I, II, III and IV).

9. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill, ISBN: 0070237530 (Unit I, II, III and IV).

10. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd, ISBN: 8175152575 (Unit I, II, III and IV).

11. Oded Goldreich, P, NP and NP - Completeness, Cambridge University Press, 2011. ISBN: 0521122546 (Unit V).

12.

MCS2C02 | Operating System Concepts

Course Number: 2.2
Contact Hours per Week: 5 (4 Lecture + 1 Practical)
Number of Credits: 4
14.
Number of Contact Hours: 60 Hrs

Prerequisite/Exposure: Advanced Data Structures (MCS1C02), Computer Organization & Architecture (MCS1C05)
Course Evaluation: 20% (Internal) + 80% (External)

Objective

- Introduce the underlying principles of an operating system
- Exposure of multi programming, virtual memory and resource management concepts
- Case study of public and commercially available operating systems

Course Outcome

The learner will be able to

- Understand the basics of operating systems
- Understand the principles of concurrency
- Compare the memory scheduling and processor scheduling algorithms
- Understand the basics of real time scheduling and client server computing

Unit I

Operating System Overview - Objectives and functions – Evolution of Operating System – Major Achievements – Process Description and Control – Process, Creation & Termination of Processes, Five State Model, Suspended Process, Process Description, Process Control – Modes of Execution, Process Creation, Process and Mode Switching. Threads – Processes Vs Threads, Multithreading, Thread States, Types of Threads, Multi Core and Multithreading. Case Study - Unix SVR4 Process Management, Linux Process and Thread Management.

Unit II

Concurrency – Principles, Race Condition, Operating System Concerns, Process Interaction, Completion for Resources, Cooperation by Sharing. Mutual Exclusion - Requirements, Hardware Support, Semaphores, Producer Consumer Problem, Monitors, Message Passing, Readers/Writers Problem. Deadlock – Principles, Prevention, Avoidance, Detection, Recovery, Dining Philosophers Problem. Case Study: Unix Concurrency Mechanisms.

Unit III

Memory Management, Address binding, Logical Vs Physical address space, Dynamic Loading, Dynamic Linking and Shared Libraries, Overlays, Swapping, Contiguous Memory allocation, Paging, Segmentation, Virtual memory, Demand paging, Page replacement, Thrashing. Case Study: Windows Memory Management.

Unit IV

Uniprocessor Scheduling – types, scheduling algorithms – criteria, nonpreemptive, preemptive. Comparative study of scheduling algorithms - FCFS, SJF, Priority, RR, Multilevel, Feedback Queue. Multiprocessor Scheduling – Classification, Granularity, Design Issues, Process Scheduling, Thread Scheduling. Real Time Scheduling - Background, Characteristics of Real Time OS, Scheduling, Deadline Scheduling, Rate Monotonic Scheduling, Priority Inversion. Case study: Linux Scheduling.

Unit V

Client/Server Computing – Definition, Applications, Classes, Three-Tier Client/Server Architecture, Middleware. Service-Oriented Architecture – Distributed Message Passing - Remote Procedure Calls - Clusters. Mobile Operating Systems – Characteristics – Comparative Study of the Features of iOS and Android.

References

1. William Stallings, Operating Systems, Internals and Design Principles, 7th Edition, Pearson, ISBN: 9780273751502.

2. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, Operating System Concepts, 9th Edition, John Wiley & Sons, ISBN: 9781118063330.

3. Ann McIver McHoes and Ida M. Flynn, Understanding Operating Systems, 6th Edition, Cengage Learning, 2010, ISBN: 9781439079201.

4. Mukesh Singhal and Niranjan G. Shivaratri, Advanced Concepts in Operating Systems – Distributed, Database, and Multiprocessor Operating Systems, Tata McGraw-Hill Education Private Limited, ISBN: 9780070575721.

5. Current Literature (for Mobile Operating Systems).

MCS2C03 | Computer Networks

Course Number: **2.3**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- To provide the student with a top down approach of networking starting from the application layer.
- To introduce computer networking in the back drop of Internet protocol stack.

Course Outcome

Student will be able to

- understand and analyze the functionality and services offered by different protocols at various layers of TCP/IP protocol stack.
- explain network security and cryptography

Course Outline

Unit I (10 hours)

Introduction to Computer networks – introduction – topology - categories of networks – Internetwork – Internet - network models - layered model - OSI and TCP/IP Models - Transmission media - Wired and unwired media. Computer networks and Internet - the network edge - the network core - network access - delay and loss - protocol layers and services – history of computer networking and Internet.

Unit II (12 hours)

Application layer protocols – principles – the web and HTTP – FTP – Email in Internet – DNS. Socket programming – building a Web server - content distribution.

Unit III (14 hours)

Transport layer services – introduction – relationship between Transport and Network layer – UDP – reliable data transfer – TCP - congestion control - Network layer services – routing – IP - routing in Internet - router - IPV6 - multicast routing.

Unit IV (12 hours)

Link layer services - error detection and correction - multiple access protocols – LAN address – ARP – Ethernet – hubs – bridges – switches - wireless links – PPP - ATM.

Unit V(12 hours)

Security in Networks – Principles of Cryptography – Authentication – Integrity – Key Distribution and Certification – Firewalls – Attacks and Counter Measures.

References

1. J. F. Kurose and K . W. Ross, *Computer Networking: A Top-Down Approach Featuring Internet*, 6th Edition, Perason Education, ISBN: 0132856204.
2. Behrouz Forouzan, *Data Communications and Networking*, 4th Edition, McGraw-Hill Reprint, ISBN: 0073250325.
3. Peterson L.L. and Davie B .S., *Computer Networks, A Systems Approach*, 5th Edition, Morgan Kaufmann, ISBN: 9780123850591.
4. Keshav, *An Engineering Approach to Computer Networking*, Pearson Education Asia, ISBN: 97898123598652000.
5. Andrew S. Tanenbaum, *Computer Networks*, 5th Edition, PHI, ISBN: 9788131787571

MCS2C04 |Artificial Intelligence

Course Number: **2.4**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs.**
Prerequisite/Exposure:
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- To introduce concepts of Artificial Intelligence and Machine Learning.

Course Outcome

The student will be able to apply the artificial intelligence and machine learning concepts in problem solving.

Course Outline

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, Best–first search, problem reduction, constraint satisfaction, means-ends analysis, heuristic in games, complexity issues.

Unit III

Knowledge representation issues, representation and mappings, representing simple facts in logic, representing instances and ISA relationships, computable functions and predicates, resolution, natural deduction, knowledge representation using rules, logic programming, forward versus backward reasoning, symbolic reasoning under uncertainty- nonmonotonic reasoning, depth first search, breadth first search.

Unit IV

Game playing – the Minimax search procedure, adding Alpha-beta cutoffs, additional refinement, iterative deepening, planning system and its components, understanding, understanding as constrained satisfaction. Slot and filler structures: semantic nets, frames, conceptual dependency, scripts. Definition and characteristics of expert system, representing and using domain knowledge, expert system shells. Knowledge engineering, knowledge acquisition, expert system life cycle & expert system tools, CYCIN & DENDRAL examples of expert system.

Unit V

Machine learning – rote learning, learning by taking advice, learning in problem solving, learning from examples, explanation based learning, analogy, formal learning theory, connectionist models - hopfield networks, learning in neural networks, back propagation, the genetic algorithm, classifier systems and genetic programming, artificial life and society based learning.

References

1. Elaine Rich, Kevin Knight and Shivshankar B. Nair, *Artificial Intelligence*, 3rd Edition, Tata – McGraw Hill, New Delhi, ISBN: 0070087709.
2. V S Janakiraman, K Sarukesi and P Gopalakrishnan, *Foundations of Artificial Intelligence and Expert System*, Macmillan India Limited, ISBN: 0333926250.
3. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, ISBN: 0136042597.
4. G. F. Luger and W.A Stubblefield, *Artificial Intelligence – Structures and Strategies for Complex Problem Solving*, Addison-Wesley, 6th Edition, ISBN: 9780321545893.
5. P. H. Winston, *Artificial Intelligence*, Addison-Wesley, 3rd Edition, ISBN: 0201533774.
6. Nils J. Nilsson, *Artificial Intelligence, A New Synthesis*, 1st Edition, Morgan Kaufmann Publishers, Inc, ISBN: 1558604677.

MCS2C04 | Principles of Software Engineering

Course Number: 2.5
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure:
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To develop familiarity with software engineering principles and practices.
- To have an understanding about the process of product/literature survey, techniques of problem definition, and methods of software development and implementation.

Course Outcome

- understand and use different software development models
- describe key activities in software development.
- explain concepts like verification and validation
- understand and apply different testing strategies

Module I: (10 Hours)

Introduction –The Software problem –Software Engineering Problem –Software Engineering Approach ;Software Process –Characteristics of a Software Process –Software Development Process –Project Management Process –Software Configuration Management Process –Process Management Process.

Module II: (14 Hours)

Software Requirements Analysis and Specification –Software Requirements –Problem Analysis –Requirements Specification –Validation –Metrics.

Module III (12 Hours)

Planning a Software Project –Cost Estimation –Project Scheduling –Staffing and Personnel Planning –Software configuration Management Plans –Quality Assurance Plans –Project Monitoring Plans –Risk Management.

Module IV (10 Hours)

Function-oriented Design –Design Principles –Module-Level Concepts–Design Notation and Specification –Structured Design –Methodology –Verification –Metrics. Detailed Design –Module specifications –Detailed Design –Verification –Metrics.

Module V (14 Hours)

Coding –Programming Practice –Top-down and Bottom-up -structured programming –Information Hiding –Programming style –Internal Documentation Verification –Code Reading –Static Analysis –Symbolic Execution –Code Inspection or Reviews –Unit Testing –Metrics –Summary Testing –

Fundamentals –Functional Testing versus structural Testing –Metrics –Reliability Estimation –Basic concepts and Definitions.

REFERENCES

1. Pankaj Jalote, “An Integrated Approach to Software Engineering”, Narosa Publishing House.
2. Pressman R.S., “Software Engineering”, Tata McGraw Hill Pub. Co.
3. Sommerville, “Software Engineering”, Pearson Education.

MCS2L02 |Practical II

Course Number: 2.6
Contact Hours per Week: 4
Number of Credits: 2
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To practically implement the theory portions covered in *Operating System Concepts (CSS2C02)* and *Computer Networks (CSS2C03).*

□ To extend the programming knowledge acquired thru *The Art of Programming Methodology (CSS1C04).*

Course Outline

Unit I: Computer Networks

Course Outcome

- design and implement simple applications based on socket programming

1. Design a LAN with a given set of requirements. The design should include topology, hardware and software requirements like cable, connectors, hubs/switches/bridges, interface cards along with a budget for the LAN. (Faculty in charge should give the requirements to the students)*.
2. Establish a LAN that consists of at least one server and two clients*.
3. Study of network utilities in Linux/Windows (hostname, ping, ifconfig, ipconfig, netstat, nslookup, telnet, traceroute, finger, telnet, tracert, arp, ftp etc)*.
4. Implementation of TCP Client.
5. Implementation of TCP Server.
6. Write a program to check the Date and Time in TCP Date Time Client.
7. Write a program to check the Date and Time in TCP Date Time Server.
8. Implementation of UDP client and server.
9. Write a program to transfer Files using UDP.
10. Implementation of transferring files using FTP.
11. Write a program to simulate the sliding window protocol.

12. Study of Network Simulators (NS2/Glomosim)*.

*These questions are NOT meant for examination purpose. However Viva questions can be asked based on these experiments.

Unit II: Operating System Concepts

1. Write programs using the following system calls: fork(), execl() and wait().
2. Write File System Calls to write, append and display.
3. To accept the burst time for a set of processes for FCFS scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.
4. To accept the burst time for a set of processes for SJF scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.
5. To accept the burst time and priority for a set of processes for Priority scheduling and create chart consisting of the burst time, priority, turnaround time and wait time of each process.
6. To create n Fibonacci numbers and prepare a list of prime numbers amongst them (use pipe for IPC).
7. To demonstrate IPC using shared memory.
8. To allocate memory requirements for processes using best fit allocation- Accept n processes with their memory requirements and n holes with their sizes. Perform memory allocation using Best Fit algorithm. Display a chart consisting of the process and the allocated hole.
9. To accept n processes with their memory requirements and n holes with their sizes. Perform memory allocation using First Fit algorithm. Display a chart consisting of the process and the allocated hole.
10. To demonstrate the process of contiguous allocation of memory blocks to store files of varying sizes.
11. To implement Producer Consumer problem using semaphores.

CSS2A02 – TERM PAPER (PROFESSIONAL COMPETENCY AUDIT COURSE)

Objectives:

- To introduce the student to the techniques of literature survey.
- To acquaint him/her with the process of presenting his/her work through seminars and technical reports.

The student is expected to do an extensive literature survey and analysis in an area related to computer science, chosen by him/her, under the supervision of a faculty member from the department. The student has to choose an area for his/her work after due consultation and approval from the guide. The study should preferably result in a critical review of the present works/design ideas /designs/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. Technical report should be prepared in TEX in IEEE conference style format.

Course Delivery Mode

Students be given choice to opt for the supervisor according to his/her area of interest. The Department council will finally decide and distribute the students among the faculty members by accommodating the choice and interest of the students, as far as possible. The faculty in charge must give proper directions and guidance to the students in carrying out the literature review effectively and systematically.

Course Evaluation & Course Credit

The Professional Competency Audit Course has 4 credits which will not be counted for evaluating the overall SGPA & CGPA. The Department shall conduct the final evaluation of the course based on the following criteria and have to intimate /upload the results of the same to the University on the stipulated date during the III Semester.

Component	Weightage
Publication of the Review Paper in a UGC Listed, Peer Reviewed or other peer reviewed refereed Journals	20% (Maximum weightage be given to UGC listed Journal and weightage be reduced in other cases)
Presentation in an International/ National/ Regional Conference	20% (Maximum weightage be given to International Conferences with Proceeding having ISBN and weightage be reduced in other cases)
Quality of the Technical Report	40%
Quality and Effectiveness of the Report Presentation	20%

Students have to obtain only minimum pass requirements in this Audit Course.

References:

Articles from ACM/IEEE/INFLIBNET Journals/Conference Proceedings and/or equivalent documents, standard textbooks and web based material, approved by the supervisor.

SEMESTER III

MCS3C01 | Advanced Database Management System

Course Number: 3.1
Contact Hours per Week: 5 (4 Lecture and 1 Practical)
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To understand the relational model, and know how to translate requirements captured in an Entity-Relationship diagram into a relational schema.
- To reason about dependencies in a relational schema.
- To understand normal form schemas, and the decomposition process by which normal forms are obtained.
- To familiarize with advanced SQL statements.
- To understand advanced features of database technologies.

Course Outcome

Student will be able to:

- Design and normalise databases for real life applications.
- Develop and deploy databases for different applications using MySQL.
- Acquire knowledge about Object Oriented and Distributed Databases.

Course Outline

Unit I

Introduction - purpose of database systems, views of data – data abstraction, instances and schemas, data independence, data models – hierarchical data model, network data model, relational data model, ER data model. Database languages - DDL, DML, transaction management, storage management, database administrator, database users, overall system structure. Relational data model - relational model concepts, keys, integrity constraints - domain constraints, key constraints, entity integrity constraints, referential integrity constraints. ER data model - basic concepts, constraints, keys, 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 and calculus - relational algebra - selection and projection, set operations, renaming, joins, division. Relational calculus - tuple relational calculus, domain relational calculus. Expressive power of algebra and calculus.

Unit II

Relational database design - anomalies in a database – functional dependency – lossless join and dependency- preserving decomposition – normalization - normal forms – first, second and third normal form – Boyce Codd normal form – multivalued, dependency – fourth normal form – join dependency – project join normal form – domain key normal form.

SQL : Data Definition,Data Manipulation, Advanced SQL, Sored Procedures and triggers.

Unit III

Transaction management, concurrency control and query processing - concept, definition and states of transactions, ACID properties – concurrency control, serializability – conflict serializability, view serializability, recoverability-recoverable schedules, non-cascading schedules, strict schedules. concurrency control schemes - locking- two phase locking.

Unit IV

Object Oriented Database Management Systems (OODBMS) - concepts, need for OODBMS, composite objects, issues in OODBMSs, advantages and disadvantages of OODBMS.

Unit V

Distributed databases - motivation - distributed database concepts, types of distribution, architecture of distributed databases, the design of distributed databases, distributed transactions, commit protocols for distributed databases. Introduction to semi structured and unstructured databases. An overview of NoSQL.

Reference

1. Elmasri and Navathe, *Fundamentals of Database Systems*, 5th Edition, Pearson, ISBN: 9788131758984.
2. Abraham Silbersehatz, Henry F. Korth and S.Sudarshan, *Database System Concepts*, 6th Edition, Tata McGraw-Hill, ISBN: 0071325220.
3. CJ Date, *An Introduction to Database Systems*, 8th Edition, Addison Wesley, ISBN: 0321197844.
4. Ramakrishnan and Gehrke, *Database Management Systems*, 3rd Edition, McGraw - Hill Education, ISBN: 9339213114.
5. Alexis Leon and Mathews Leon, *Database Management Systems*, 1st Edition, Vikas Publishers, ISBN: 8182092221.
6. Vikram Vaswani, *MySQL The complete Reference*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070586845.
7. Joel Murach, *Murach's Mysql*, Mike Murach & Associates Inc, ISBN: 9350237

8. Guy Harrison, *Next Generation Data Bases – NoSQL, NewSQL and Big Data*, 1stEd ,Apress, 2015.

MCS3C02 | Principles of Compilers

Course Number: 3.2
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: Advanced Data Structures (MCS1C02), Operating System Concepts (MCS2C02)
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To introduce the fundamental concepts and various phases of compiler design.

Course Outcome

Unit I

Introduction to compiling - definition of compiler, translator, interpreter, analysis of the source program, the phases of a compiler, grouping of phases, compiler construction tools - applications of compiler technology – lexical analysis – role of lexical analyzer – input buffering - specification of tokens.

Unit II

Syntax analysis – role of parser – error handling and recovery – definitions of parsing, top-down parsing and bottom-up parsing - context free grammars – derivations - parse tree – top-down parser – recursive descent parsing - FIRST and FOLLOW – LL (1) Grammars – recursive predictive parsing - bottom up parser – reductions – handle pruning – shift reduce parsing - operator precedence parsing.

Unit III

Intermediate code generation – Overview of intermediate code forms: Syntax tree, DAG, three address code, quadruples, triples, Static Simple Assignment form. Declarations – Translation of Expressions. Case study: Translation of if, if else and while statements.

Unit IV

Code generation – issues in the design of a code generator – the target language – a simple target machine model – the program and instruction costs – address in the target code – static allocation – stack allocation – run-time address for names – basic blocks and flow graphs – representation of flow graphs.

Unit V

Code optimization - the principal sources of optimization – data flow analysis – abstraction – data flow analysis schema – data flow schemas on basic blocks – reaching definitions – live variable analysis – available expressions. Region based analysis – regions – region hierarchies for reducible flow graphs – overview of a region based analysis.

References

1. V Aho A, Ravi Sethi, D Ullman J, Compilers Principles, Techniques and Tools, 2nd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131721019.

2. K. V. N. Sunitha, Compiler Construction, Pearson, ISBN:9789332500297.
3. W Appel and Andrew, Modern Compiler Implementation in C, 1st Edition, Cambridge University Press, ISBN: 817596071X.
4. Allen I Holub, Compiler Design in C, 1st Edition, PHI Learning Pvt Ltd, ISBN: 812030778X.
5. Tremblay and Sorenson, The Theory and Practice of Compiler Writing, 1st Edition, BSP Books Pvt Ltd, ISBN: 8178000776.
6. Torben Ægidius Mogensen, Basics of Compiler Design, Department of Computer Science, University of Copenhagen (Online Edition).

CS3C03 | Object Oriented Programming Concepts

7.

Course Number: **3.3**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **The Art of Programming Methodology (MCS1C04)**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- Understand object oriented concepts and methodologies and to learn its implementation using Java.
- Learn modelling diagram and its use in software development

Prerequisites

- Knowledge of any programming language.

Course Outcome

- Choose the appropriate design methodologies for a specific problem.
- Design and develop applications in java programming language.
- Develop sophisticated, interactive user interfaces using the Java Swing class and appropriate layout managers.

Unit I (12 Hours)

Introduction to OOPS - basic principles of object orientation (objects , attributes and methods, encapsulation and information hiding, state retention, object identity, messages, class hierarchy, inheritance, polymorphism, genericity) - introduction to Java - history, versioning, the Java Virtual Machine, byte code, features of Java, language components - primitive data types, comments, keywords, literals, variables scope & declarations, control structures - FOR, IF, WHILE, DO WHILE, SWITCH, BREAK, CONTINUE statements - operators - casts and conversions - arrays.

Unit II (12 Hours)

Object - oriented programming – classes - class fundamentals - declaring objects - new operator – methods – parameter passing – constructors – parameterized constructors - this keyword – finalize method. Overloading methods and constructors, access controls, static and final, nested and inner classes. Inheritance - extends, member access and inheritance, super keyword, polymorphism, method overriding, dynamic method dispatch, abstract classes, packages and interfaces.

Unit III (12 Hours)

Exceptions, threads & IO in Java - The file and standard streams, stream classes and interfaces, using byte streams and character streams, threads - threads vs. processes, creating threads, runnable interface, thread class, inter thread communication, synchronization. Exceptions - basic of Java exception handling, hierarchy, developing user defined exception classes.

Unit IV (12 Hours)

Database, sockets Swing – JDBC - introduction, architecture, drivers, connections, statements, resultset and meta data. Sockets: introduction to networking, InetAddress, url, socket, server sockets. Swings - Japplets, icon, labels, buttons, textbox, combo box, tables.

Unit V (12 Hours)

Java Server Pages (JSP) - JSP, life cycle, scripting elements, implicit objects-out, request, response, directive elements, action element - jsp:include, create and use JSP error pages.
Introduction to Unified Modelling Language (UML), UML diagrams, class diagrams, object interaction diagrams, state and activity diagrams, component diagrams, deployment diagrams.

References

1. Herbert Scheldt, Java Complete Reference, 8th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 1259002462.
2. E Balaguruswamy, Programming in Java, : A Primer, 4th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 007014169X.
3. Phil Hanna , JSP 2.0: The Complete Reference
4. David Flanagan, Jim Farley, William Crawford and Kris Magnusson, Java Enterprise in a Nutshell: A Desktop Quick Reference, 3rd Edition, O'Reilly Media, ISBN: 0596101422.
5. Grady Booch, James Rumbaugh and Ivar Jacobson, The Unified Modeling Language User Guide, 2nd Edition, Pearson, ISBN: 8131715825.

MCS3L03 | Practical III

Course Number: 3.6
Contact Hours per Week: 4
Number of Credits: 2
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To practically implement the theoretical aspects covered in *Advanced Database Management System (MCS3C01)* and *Object Oriented Programming Concepts (MCS3C03).*
- To extend the programming knowledge acquired through *The Art of Programming Methodology (MCS1C04)* to encompass object oriented techniques.

Course Outcome

Student will be able to develop programs by incorporating their knowledge in *Database Management System (MCS3C01)* and *Object Oriented Programming Concepts (MCS3C03).*

Course Outline

Unit I: Advanced Database Management System

Data definition in SQL
- data types, creation, insertion, viewing, updation, deletion of tables, modifying the structure of the tables, renaming, dropping of tables. Data constraints – I/O constraints, primary key, foreign key, unique key constraints, ALTER TABLE command –
database manipulation in SQL - computations done on table data - SELECT command, logical operators, range searching, pattern matching, grouping data from tables in SQL, GROUP BY, HAVING clauses. Joins – joining multiple tables, joining a table to it. DELETE – UPDATE. Views - creation, renaming the column of a view, destroys view

Program with SQL - data types Using SET and SELECT commands, procedural flow, IF, IF /ELSE, WHILE, GOTO, global variables. developing stored procedures, CREATE, ALTER and DROP, passing and returning data to stored procedures, using stored procedures within queries, building user defined functions, creating and calling a scalar function, implementing triggers, creating triggers, multiple trigger interaction (Use MySQL as the RDBMS).

1. Creating database tables and using data types (create table, modify table, drop table).
2. Data Manipulation (adding data with INSERT, modify data with UPDATE, deleting records with DELETE).
3. Implementing the Constraints (NULL and NOT NULL, primary key and foreign key constraint, unique, check and default constraint).
4. Retrieving Data Using SELECT (simple SELECT, WHERE, IN, BETWEEN, ORDERED BY, DISTINCT and GROUP BY).
5. Aggregate Functions (AVG, COUNT, MAX, MIN, SUM).
6. String functions.
7. Date and Time Functions.
8. Use of union, intersection, set difference.
9. Implement Nested Queries & JOIN operation.
10. Performing different operations on a view.
11. Stored Procedure Programming – Simple Procedures – decision making – Loops – Error handlers – Cursors – Functions - Triggers – Calling Stored Procedure from Triggers.

Unit II: Object Oriented Programming Concepts

1. Simple Java programs like computing formulas expressions.
2. Programs involving loops and decisions like generating Fibonacci, prime, strange series.
3. Programs involving arrays.
4. Programs involving class and objects.
5. Illustrate method overloading.
6. Illustrate single level inheritance.
7. Illustrate multiple inheritances using interface.
8. String sorting, pattern matching etc.
9. Illustrate threads and thread priorities.
10. Illustrate the use of Packages.
11. Exception handling (user-defined).
12. Abstract class.
13. Method overriding.
14. Illustrate usage of Applets like moving ball, face etc.
15. Create an AWT application for a simple calculator.
16. Frame application to illustrate the window events.
17. Frame application to illustrate mouse and keyboard event handling.
18. Swing applications.
19. Create a JDBC application to add the details of a student into a table (Use MySQL as the RDBMS).
20. Socket Programming.

Elective I MCS3E01| List of Electives

MCS3E01a | Computer Graphics

Course Number: **3.4a**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- □ To understand the fundamentals of the modern computer graphics.
- □ To pipeline the mathematics of affine transformations in three dimensions.
- □ To understand the common data structures to represent and manipulate geometry, colour and light representation and manipulation in graphics systems.
- □ To have an exposure to programming in Open GL.

Course Outcome

Students will be able to use different concepts in computer graphics implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.

Course Outline

Unit I

Introduction – Application of computer graphics, Video Display Devices- refresh CRT, raster and random scan display, color CRT, flat panel, LCD, LED, DVST. Raster -Scan Systems-video controller, display processor, Random-Scan Systems.

Unit II

2D Graphics: Line drawing algorithms – DDA, Bresenham's – Midpoint Circle drawing algorithm – Filling-Scan line polygon fill algorithm, boundary fill algorithm, floodfill algorithm, 2D Transformations-translation, rotation, scaling, shearing and reflection, composite transformations. 2D Viewing –the viewing pipeline, viewing coordinate reference frame, window-to- viewport coordinate transformation. Clipping-point clipping, Cohen Sutherland line clipping, Sutherland Hodgeman polygon clipping, text clipping.

Unit III

3D Graphics: 3D Transformations- translation, rotation, scaling, shearing and reflection,3D Viewing-viewing pipeline, viewing coordinates, projections- parallel & perspective projections.

Unit IV

3D object representation - wireframe model, curve representation, surfaces, spline representation, bezier curves, cubic spline. Visible surface detection methods-classification, back-face detection, Z-buffer algorithm.

Unit V

Discrete Techniques and OpenGL programming - Texture mapping, Bit and Pixel operations, Compositing, Sampling and Aliasing Techniques. Introduction to OpenGL, Features in OpenGL, OpenGL operations, Abstractions in OpenGL – GL, GLU & GLUT, a few examples of OpenGL programs.

References

1. Donald Hearn and M. Pauline Baker, *Computer Graphics*, 2nd Edition, Prentice Hall, ISBN: 0135309247.

2. Donald D. Hearn, M. Pauline Baker and Warren Carithers, *Computer Graphics with Open GL*, 4th Edition, Prentice Hall, ISBN: 9780136053583.
3. Hill, *Computer Graphics using OpenG,L*, 3rd Edition, Prentice Hall of India Private Ltd. New Delhi, ISBN: 8120338294.
4. Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, Dave Shriner and Tom David, *Open GL Programming Guide*, 6th Edition, Person, ISBN: 9780201604580.
5. The Official Guide to Learning OpenGL, Version 1.1, Available at http://www.glprogramming.com/red/.
6. Shreiner and Angel, *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*, 6th Edition, Pearson Education, ISBN: 0132545233.

MCS3E01b | Introduction to Soft Computing

Course Number: **3.4b**

Contact Hours per Week: **4**

Number of Credits: **4**

Number of Contact Hours: **60 Hrs**

Prerequisite/Exposure: **None**

Course Evaluation: **25% (Internal) + 75% (External)**

Objectives

□ To give students the fundamental knowledge of soft computing theories.

□ To expose the fundamentals of non-traditional technologies and approaches to solving hard real-world problems.

Course Outcome

Students will be able to

- understand and apply fundamental knowledge of soft computing.
- introduce genetic algorithm and its applications in optimization.
- familiarize with neural networks and learning methods for neural networks.

Course Outline

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 - decision surfaces – error probabilities and integrals - normal density - discriminant functions for normal density.

Unit II

Introduction to genetic algorithm, genetic operators and parameters, genetic algorithms in problem solving, theoretical foundations of genetic algorithms, implementation issues – systems.

Unit III

Neural model and network architectures, perceptron learning, supervised hebbian learning, back-propagation, associative learning, competitive networks, hopfield network, computing with neural nets and applications of neural network.

Unit IV

Introduction to fuzzy sets, operations on fuzzy sets, fuzzy relations, fuzzy measures, applications of fuzzy set theory to different branches of science and engineering.

Unit V

Advanced topics - support vector machines, evolutionary computation (EC) – evolutionary algorithms, harmony search, swarm intelligence.

References

1. Chuen-Tsai Sun, Eiji Mizutani and Jyh-Shing Roger Jang, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall India, ISBN: 8120322436.
2. M. Mitchell, *An Introduction to Genetic Algorithms*, Prentice-Hall, ISBN: 0262631857.
3. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, ISBN: 0785342157673.
4. S. V. Kartalopoulos, *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*, Wiley-IEEE Press, 1st Edition, ISBN: 07803112802004.
5. S. Rajasekaran and G. A. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications*, PHI, ISBN: 9788120321861.

MCS3E01c | Web Technology

Course Number: 3.4c
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To introduce the tools for creating and maintaining websites – content development (HTML), client side scripting (JavaScript), web server (Apache), server side scripting (PHP) and content management system (Joomla!).

Course Outcome

Students will be able to

develop and maintain websites by utilizing their knowledge in content development (HTML), client side scripting (JavaScript), web server (Apache), server side scripting (PHP) and content management system (Joomla!).

Course Outline

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. Overview of HTML - basic formatting tags - heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image. Attributes - align, color, bgcolor, font face, border, size. Navigation links using anchor tag - internal, external, mail and image links. Lists - ordered, unordered and definition, table tag, HTML form controls - form, text, password, textarea, button, checkbox, radio button, select box, hidden controls, frameset and frames. CSS.

Unit II

Client side programming – introduction – popular client side scripting languages - Java Script - introduction, identifiers, operators, functions, event handling, classes, objects, array, math, string, window object, navigator DHTML font, text, image change, table expansion. JavaScript's object model-strengths and weaknesses of JavaScript - building and extending objects in JavaScript - events in JavaScript - event handlers - creating interactive forms – cookies - storing users choices in cookies - encoding cookies - browser objects - object hierarchy, creating browser objects, working with window, document, history & location - browser detection, Java to JavaScript communication.

Unit III

Web server – role - Apache web server – introduction – architecture – features - Apache's role in the Internet – LAMP – WAMP - installation and configuration - build and install Apache web server - verify initial configuration start, stop, and status the Apache server process. Configure Apache core modules security - basic security with Apache - host-based authentication - user-based authentication - secure sockets layer (SSL) - delivering dynamic web content - Apache's role in the dynamic web - server side includes (SSIs) - configure Apache web server to support CGI – CGI Alternative Technologies. virtual hosts, redirection, indexing – virtual hosting with Apache, virtual host configuration redirection, directory indexing. Proxy servers and firewalls - apache proxy configuring, proxy services firewalls and apache, firewall architecture models monitoring apache web server - error logs, logging http access, web server status and server information, user tracking - proxy caching.

Unit IV

Server side programming – server side scripts – PHP – designing dynamic web pages using PHP - defining PHP variables – variable types – operators – control flow constructs in PHP – passing form data between pages - establishing connection with MySQL database – managing database.

Unit V

Overview of content management system - coding for reusability (header.php) – user management - article publishing - additional CMS features – Web site development using Joomla!.

References

1. Thomas A. Powell, *The Complete Reference HTML,* 3rd Edition, McGraw-Hill/Osborne Media, ISBN: 0072129514.
2. Thomas A. Powell, *Web Design: The Complete Reference,* 2nd Sub-Edition, McGraw-Hill/Osborne Media, ISBN: 0072119772
3. Robert W. Sebesta, *Programming with World Wide Web,* 7th Edition, Addison-Wesley, ISBN: 9780132665810.
4. Xue Bai, Michael Ekedahl, Joyce Farrell, Don Gosselin, Diane Zak, Bill Morrissey, Michael V. Ekedahl, Peter Macintyre and Shashi Kaparthi, *The Web Warrior Guide to Web programming,* Thomson Learning, ISBN: 9780619064587.
5. Chris Bates, *Web Programming: Building Internet Applications*, 3rd Edition, Wiley Academic Catalog, ISBN: 9780470017753.
6. Paul J. Deitel, Harvey M. Deitel, Harvey Deitel, Paul Deitel and Abbey Deitel, *Internet and World Wide Web: How to Program*, 5th Edition, Prentice Hall, ISBN: 9780132151009.
7. R. Allen Wyke and Richard Wagner, *JavaScript Unleashed*, 3rd Edition, SAMS, ISBN: 9780672317637.
8. Richard Bowen Ken Coar, Ken A Coar and Matthew Marlowe, *Apache Server Unleashed*, SAMS, ISBN: 0672318083.
9. Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K Glass, *Beginning PHP5, Apache, and MySQL Web Development*, Wrox, ISBN: 0764579665.
10. Jennifer Marriott and Elin Waring, *The Official Joomla! Book*, Addison-Wesley Professional, ISBN: 978-0321821546.
11. Ron Severdia and Kenneth Crowder, *Using Joomla: Building Powerful and Efficient Web Sites*, 1st Edition, O'Reilly Media, ISBN: 9780596804947.

MCS3E01d | Bioinformatics

Course Number: 3.4d
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- Expose students to the popular genomic and proteomic databases and to impart knowledge in processing and analyzing genomic data.
- Introduce advanced topics in Bioinformatics.

Course Outcome

Students will

- get knowledge about basic terms and concepts of bioinformatics.
- familiarize pair wise sequence alignment and multiple sequence alignment methods.
- able to process and analyse data taken from popular genomic and proteomic databases

Course Outline

Unit I

Bioinformatics - introduction to - nature and scope of computational biology and Bioinformatics. Cells - prokaryotes and eukaryotes - DNA double helix - central dogma – RNA, Amino acids, Proteins - string representations. A glossary of Bioinformatics terms - file format for bio-molecular sequences, sequence alignment, phylogeny, gene finding, microarray analysis, homology and evolutionary relationships.

Unit II

Basic algorithms in Computational Biology - exhaustive search methods and their applications in Computational Biology - string matching algorithms. Motif finding - tandem repeats – concept of dynamic programming - graph algorithms - clustering algorithms.

Unit III

Sequence alignment - pair-wise sequence alignment, need of scoring schemes - penalizing gaps, scoring matrices for amino acid sequence alignment, PAM probability matrix and log odds matrix, BLOSUM, Dot-plot visualization, Needleman-Wunsch algorithm- effect of scoring schemes – evalues - BLAST and FASTA, Smith – Waterman algorithm for local alignment.

Unit IV

Multiple sequence alignment - sequence alignment using dynamic programming, N-dimensional dynamic programming. Tools for MSA - muscle and T-Coffee. Phylogenetic algorithms - evaluation of phylogenetic trees, significance.

Unit V

Introduction to the major resources - 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 – procedures to access these databases and to make use of the tools available.

Text Books

1. Mount D, Bioinformatics*: Sequence & Genome Analysis,* 2nd Edition, Cold spring Harbor Press, ISBN: 978-087969712.
2. Dan Gusfield, *Algorithms on Strings Trees and Sequences,* 1st Edition, Cambridge University Press, ISBN: 0521585198.
3. Pevzner P A, *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, Cambridge, MA, ISBN: ISBN: 9780262161978.
4. Jeremy J. Ramsden, *Bioinformatics: An Introduction*, Springer, ISBN: 9789401570961.
5. Sushmita M and Tinku A, *Data Mining: Multimedia, Soft Computing and Bioinformatics*, Wiley-Interscience, ISBN: 9780471460541.

References

1. Richard M. Karp, *Mathematical Challenges from Genomics and Molecular Biology*, Notices of the American Mathematical Society, vol. 49, no. 5, pp. 544-553.
2. Glyn Moody, *Digital Code of Life: How Bioinformatics is Revolutionizing*

Science, Medicine and Business, ISBN: 9780471327882.

3. Tao Jiang, Ying Xu and Michael Q. Zhang, *Current Topics in Computational Molecular Biology Edible Oil Processing*, 1st Edition, Ane Books Pvt Ltd, ISBN: 9788180520525.
4. Andrzej K. Konopka and M. James C. Crabbe, *Compact Handbook of Computational Biology*, 1st Edition, CRC Press, ISBN: 9780824709822.
5. Richard E. Bellman, *Dynamic Programming*, Princeton University Press, ISBN: 9780691146683.
6. Needleman S B and Wunsch C D, *A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins*, J. Mol. Biol., 48 (1970) 443–453.
7. Smith T F and Waterman M S, *Identification of Common Molecular Subsequences*, J. Mol. Bio. 147 (1981) 195–197.
8. Watson J D and Crick F H C, *A Structure for Deoxyribose Nucleic Acid*, Nature, 171 (1953) 737–738.
9. Pevzner P A and Waterman M S, *Open Combinatorial Problems in Computational Molecular Biology*, Proc. Third Israel Symp. Theo. Comp. Syst. IEEE Computer Society Press, (1995) 158 – 173.

MCS3E01e |Computer Optimization Techniques

Course Number: 3.4e
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To give an exposure for the student to the area of modeling techniques, numerical methods and algorithms.

□ To realize the importance of various aspects of optimization techniques in industries like IT.

□ To implement the knowledge of optimization techniques in real life problems.

Course Outcome

Student will

- understand and apply different modeling techniques.
- get knowledge to implement optimization techniques in real life problems.

Course Outline

Unit I

Linear programming and sensitivity analysis – two variable LP model, graphical and algebraic LP solutions, some LP applications, the simplex method and sensitivity analysis, primal-dual relationships and economic interpretation, dual simplex and generalized simplex algorithms and post-optimal analysis.

Unit II

Transportation and Network models - The transportation models and algorithm, the assignment and transshipment models, minimum spanning tree algorithm, shortest-route problem, maximum flow and min-cost models, critical path method and algorithms for matching.

Unit III

Advanced linear programming and applications - simplex method fundamentals, revised simplex method and computational considerations, bounded variables algorithm, duality, parametric linear programming, goal programming formulations and algorithms.

Unit IV

Integer linear programming - illustrative applications, integer programming algorithms, unimodularity and cutting-plane methods, travelling salesperson problem.

Unit V

Dynamic programming (DP) and its application - recursive nature of computations in DP, forward and backward recursion, selected DP applications, problem of dimensionality, branch and bound method and dynamic programming, some deterministic inventory models. Nonlinear programming - convex programming problems, unconstrained problems and algorithms, constrained problems and algorithms.

References

1. H. A. Taha, *Operations Research: An Introduction*, 9th Edition, Pearson Prentice Hall, ISBN: 013255593X.
2. C. H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, ISBN: 9780486402581.

MCS3E01f |Numerical and Statistical Methods

Course Number: 3.4f
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To provide the student with basic concepts in statistics, probability that can be applied for mathematical modeling of computer applications.

Course Outcome

Students will

get knowledge to apply basic concepts in statistics for mathematical modeling of computer applications**.**

Course Outline

Unit I

Approximation and errors in computing - introduction, significant digits - inherent errors – numerical error - modeling errors - blunders - absolute and relative errors - conditioning and stability. Roots of non-linear equations - introduction - iterative methods – bisection - false position – Newton - Raphson’s, Secant and Bairstow’s methods.

Unit II

Introduction solution of linear equations - Gauss elimination - Gauss-Jordan method
- Jacobi Iteration method - Gauss-Seidal methods. Interpolation - linear interpolation - Newton’s forward backward & divided difference interpolation methods – Lagrange’s method.

Unit III

Integration - trapezoidal rule, Simpson’s 1/3, & 3/8 rules. Differential equations: Heunn’s polygon, Range- Kutta fourth order, Milne-Simpson, Adams-Bashforth and Adams-Moulton methods.

Unit IV

Classical definition of probability – statistical definition of probability – axiomatic approach to probability – addition and multiplication theorem on probability - compound and conditional probability – independence of events – Bayes theorem random variables – discrete and continues – pmf, pdf and distribution functions.

Unit V

Introduction linear programming – mathematical formulation – graphical method of solution – simplex method – duality – dual simplex – transportation – assignment problems.

References

1. E. Balagurusamy, *Numerical Methods*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0074633112.
2. S.G. Gupta and V.K. Kapoor, *Fundamentals of Mathematical Statistics*, 11th Edition, Sultan Chand & Sons , ISBN: 9788180545283.
3. V.Rajaraman, *Computer Oriented Numerical Methods*, 3rd Edition, Prentice Hall Of India, ISBN: 81203078601993.
4. Satyendra Mittal and C. P. Sethi, *Linear Programming*, Pragati Prakashan.

Elective II MCS3E02 | List of Electives

MCS3E02a | Pattern Recognition

Course Number: **3.5a**
Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- To understand the concept of a pattern and the basic approach to the development of pattern recognition algorithms.
- To understand and apply methods for preprocessing, feature extraction, and feature selection to multivariate data.
- To understand supervised and unsupervised classification methods to detect and characterize patterns in real-world data.

Course Outcome

Student will get knowledge about the concept of patterns and will be able to apply different pattern recognition algorithms.

Course Outline

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 - 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 technique – 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 – nonseparable behavior - 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, and 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. Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern Classification*, CBS Publishers & Distributors, 2nd Edition, ISBN: 9788126511167.
2. Gonzalez R.C. and Thomson M.G., *Syntactic Pattern Recognition: An Introduction*, 1st Edition, Addison-Wesley, ISBN: 0201029316.
3. Fu K. S., *Syntactic Pattern Recognition and Applications*, Prentice Hall, ISBN: 0138801207.
4. Rajjan Shinghal, *Pattern Recognition: Techniques and Applications*, 1st Edition, Oxford University Press India, ISBN: 0195676858.

MCS3E02b | Wireless & Mobile Networks

Course Number: 3.5b
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: Computer Networks (CSS2C03), Object Oriented Programming Concepts (CSS3C03)
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To understand the fundamental concepts of wireless and mobile networks.
□ To familiarize with wireless application Protocols to develop mobile content applications.
□ To understand about the security aspects of wireless networks.
□ To learn programming in the wireless mobile environment.

Course Outcome

Students will

- understand and apply wireless application protocol to develop mobile content application.
- be able to track security aspects of wireless networks

Course Outline

Unit I

Introduction - applications - brief history of wireless communication – open research problems – wireless transmission – frequencies for radio transmission – signals – antennas – signal propagation – multiplexing – modulation – spread spectrum – cellular systems – medium access control – motivation – SDMA – FDMA – TDMA – CDMA – comparison.

Unit II

Different generations of Wireless Cellular Networks - 1G, 2G, 2.5G, 3G, 4G. Telecommunication systems – GSM – DECT – TETRA – UMTS – IMT-2000. Wireless LAN – Infrared Vs. Radio transmission – Infrastructure Vs. Adhoc networks – IEEE 802.11 – HIPERLAN – Bluetooth.

Unit III

Mobile network layer - Mobile IP – Dynamic Host Configuration Protocol - Routing – DSDV – DSR – Alternative Metrics. Transport and application layers - traditional TCP – classical TCP improvements – WAP, WAP 2.0.

Unit IV

Wireless network security – IEEE 80211i security – Wireless Transport Layer Security – sessions and connections – protocol architecture – WAP end-to-end security.

Unit V

Java for wireless devices - setting up the development environment - basic data types, libraries (CLDC, MIDP) - UI controls - displayable and display image - events and event handling - list and choice - text box - alerts - persistent storage - record stores – records - record enumeration - network MIDlets - the connection framework - connection interface - connection using HTTP - datagram connection.

References

1. Jochen Schiller, *Mobile Communications*, Pearson Education, 2nd Edition, ISBN: 8131724263.
2. Raj Kamal, *Mobile Computing*, 2nd Edition Oxford Univ Press, ISBN: 0198068913.
3. William Stallings, *Network Security Essentials Applications and Standards*, 4th Edition, Pearson India, ISBN: 8131761754.
4. Yu Feng and Jun Zhu, *Wireless Java Programming with J2ME*, 1st Edition, Sams, ISBN: 0672321351.
5. Dreamtech Software Team, *Wireless Programming with J2ME: Cracking the Code*, Wiley, ISBN: 0764548859.
6. William Stallings, *Wireless Communications and Networks*, 2nd Edition, Pearson India, ISBN: 8131720934.
7. Jochen Burkhardt, Horst Henn, Stefan Hepper, Klaus Rindtorff and Thomas Schaeck, *Pervasive Computing Technology and Architecture of Mobile Internet Applications*, 14th Edition, Pearson Education, ISBN: 8177582801.
8. Nishit Narang and Sumit Kasera, *2G Mobile Networks: GSM and HSCSD*, Tata McGraw Hill Education, ISBN: 0070621063.
9. Hasan Ahmed, Roopa Yavagal and Asoke K Talukder, *Mobile Computing: Technology, Applications and Service Creation*, 2nd Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070144575.

MCS3E02c | Cryptography and Network Security

Course Number: 3.5c
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure:
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To be familiar with classical and modern encryption and decryption techniques and apply in the security system.
- To understand various aspects of network security standards.

Course Outcome

Students will be able to apply classical and modern encryption and decryption techniques in security system.

Course Outline

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.

MCS3E02d| Advanced Web Technology

Course Number: 3.5d

Contact Hours per Week: 4

Number of Credits: 4

Number of Contact Hours: 60 Hrs

Prerequisite/Exposure: None

Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To introduce the advanced concepts of web development tools – Web 2.0, Web Services, Python, SQLite and MVC architecture.

Course Outcome

Student will be able to develop web applications utlising the advanced concepts of web development tools – Web 2.0, Web Services, Python, SQLite and MVC architecture.

Course Outline

Unit I

Web 2.0 - definition, characteristics, key features, client side technologies (Ajax and JavaScript frameworks - YUI library, Dojo toolkit, MooTools, jQuery, Ext JS and prototype JavaScript framework), server side technologies (Ruby, Perl, Python, Enterprise Java J2EE and Microsoft.NET Framework), concepts (Rich Internet Application — Web-Oriented Architecture — Social Web), SLATES.

Unit II

Fundamentals of Web Services - Definition, Components, benefits, behavioral characteristics. Web services architecture - web service roles, web service protocol stack, service transport. web services components - XML-RPC, SOAP, WSDL, UDDI. web services security (notions) - confidentiality (XML-RPC and SOAP run on top of HTTP) - support for Secure Sockets Layer (SSL) for HTTP - encrypted communication via SSL, authentication (HTTP's built-in support for Basic and Digest authentication - SOAP security extensions - Digital Signature – SOAP - DSIG - SAML).

Unit III

Introduction to Python – installation – Python interpreter – usage and customization – editor setup – variables, expressions and statements – functions. Strings – lists – list comprehensions – stacks – queues – tuples – sequences – sets – dictionaries – sets - modules, I/O and exception handling - modules – search path – compiled modules – standard modules – packages – input and output functions – files – read and write – exception – handling and raising – user defined exceptions.

Unit IV

Server side programming using Python - server side scripting - CGI - role of Web server – Apache web server – Python server side script – developing Python Server Side Pages (PSP) – capturing form data – validation – processing data – exchange of data between form and server.

Unit V

Python-SQLite integration - features of SQLite, data types, introduction to SQL commands - SELECT, DELETE, UPDATE, INSERT. Python functions for SQLite operations – database connection, database and table creation, selection, query, fetching results - insertion and deletion of data using Python - displaying data from SQLite in webpage. Case study - server MVC design pattern – Django.

References

1. James Governor, *Web 2.0 Architectures : What Entrepreneurs & Information Architects Need to Know*, 1st Edition, Shroff Publisher & Distributors, ISBN: 8184047355.
2. S. V. Subrahmanya and B. V. Kumar, *Web Services: An Introduction*, 2nd Edition, Tata Mc-graw Hill Publishing Co. Ltd, ISBN: 1259002764.
3. Web 2.0, http://en.wikipedia.org/wiki/Web_2.0
4. Web Services, http://www.tutorialspoint.com/webservices/
5. Ron Schmelzer, Michael Qualls, Sam Hunting, David Houlding, Madhu Siddalingaiah, Jason Bloomberg, Travis Vandersypen, Chad Darby and Diane Kennedy, *XML and Web Services Unleashed*, Sams, ISBN: 0672323419.
6. Sandeep Chatterjee, James Webber, *Developing Enterprise Web Services: An*

Architect's Guide, 1st Edition, Pearson India, ISBN: 8131713172.

7. The Python Tutorial, http://docs.python.org/3.3/tutorial/
8. Allen Downey, Jeffrey Elkner and Chris Meyers, *How to Think Like a Computer Scientist: Learning with Python, Createspace, 2009,* ISBN: 1441419071. Online Version: http://openbookproject.net/thinkcs/python/english3e/
9. Python Documentation. Available at http://www.python.org/doc/
10. Swarooph CH, *A Byte of Python*. Available at http://swaroopch.com/notes/python/
11. Wesley J Chun, *Core Python Programming*, 2nd Edition, Pearson Education, ISBN: 8131711889.

MCS3E02e| Virtualisation and Cloud Computing

Course Number: 3.5e
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- Understand the technical capabilities and business benefits of virtualization and cloud computing and how to measure these benefits.
- Describe the landscape of different types of virtualization and understand the different types of clouds.
- Illustrate how key application features can be delivered on virtual infrastructures.
- Explain typical steps that lead to the successful adoption of virtualization technologies.

Course Outcome

Student will

- get knowledge about different types of clouds and different types of virtualization.
- learn to adopt virtualization technologies.

Course Outline

Unit I

Introduction - evolution of cloud computing – system models for distributed and cloud computing – NIST cloud computing reference architecture – Infrastructure as a Service (IaaS) – resource virtualization – Platform as a Service (PaaS) – cloud platform & management – Software as a Service (SaaS) – available service providers.

Unit II

Virtualization - basics of virtualization - types of virtualization - implementation levels of virtualization - virtualization structures - tools and mechanisms - virtualization of CPU, memory, I/O devices - desktop virtualization – server virtualization – Linux KVM, Xen, Qemu, LXC, OpenVZ.

Unit III

Cloud infrastructure - FOSS cloud software environments - Eucalyptus, Open Nebula, OpenStack – OpenStack architecture – compute, object storage, image service, identity, dashboard, networking, block storage, metering, basic cloud orchestration and service definition.

Unit IV

Programming model - parallel and distributed programming paradigms – Mapreduce, twister and iterative Mapreduce – mapping applications - programming support – Apache Hadoop – HDFS, Hadoop I/O, Hadoop configuration, MapReduce on Hadoop.

Unit V

Security in the cloud - security overview – cloud security challenges – software-as- a-service security – security governance – risk management – security monitoring – security architecture design – data security – application security – virtual machine security – Qubes – desktop security through Virtualization.

References

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, *Distributed and Cloud Computing (From Parallel Processing to the Internet of Things)*, Elsevier Science, ISBN: 9780128002049.
2. John W. Rittinghouse and James F. Ransome, *Cloud Computing: Implementation, Management, and Security*, 1st Edition, CRC Press, ISBN: 1439806802.
3. Toby Velte, Robert Elsenpeter and Anthony Velte, *Cloud Computing, A Practical Approach*, TMH, ISBN: 9780071626958.
4. George Reese, *Cloud Application Architectures*, 1st Edition, Shroff /O'Reilly, ISBN: 8184047142.
5. Ravi Nair and Jim Smith, *Virtual Machines: Versatile Platforms for Systems and Processes*, 1st Edition, Elsevier Science / Morgan Kaufmann, ISBN: 9780080525402/ 1558609105.
6. Katarina Stanoevska - Slabeva, Thomas Wozniak, Santi Ristol, *Grid and Cloud Computing – A Business Perspective on Technology and Applications*, Springer, ISBN: 3642051928.
7. Open stack Operations Guide, http://docs.openstack.org/ops/.
8. Tom White, *Hadoop: The Definitive Guide*, O'Reilly Media, ISBN: 9780596551360.

MCS3E02f | Data Warehousing and Data Mining

Course Number: 3.5f
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

□ To provide the fundamentals on information retrieval and data mining techniques

□ To focus on practical algorithms of textual document indexing, relevance ranking, web usage mining, text analytics, as well as their performance evaluations.

□ To give an exposure to the fundamentals of Data Analytics.

Course Outcome

Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on large data sets.

Course Outline

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, ISBN: 9380931913.
2. Alex Berson and Stephen J. Smith, *Data Warehousing, Data Mining & OLAP*, Computing Mcgraw-Hill, ISBN: 0070062722.
3. K.P. Soman, Shyam Diwakar and V. Ajay, *Insight into Data mining Theory and Practice*, 1st Edition, Prentice Hall of India, ISBN: 8120328973.
4. G. K. Gupta, *Introduction to Data Mining with Case Studies*, 3rd Edition, PHI Learning Pvt. Ltd, ISBN: 8120350022.
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, *Introduction to Data Mining*, 1st Edition, Pearson India, ISBN: 9332518653.

SEMESTER IV

Elective III MCS4E03 | List of Electives

MCS4E03a | Data Compression

Course Number: **4.1a**
Contact Hours per Week: **5**
Number of Credits: **3**
Number of Contact Hours: **75 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- To understand the physical significance of some basic concepts of information theory including entropy, average mutual information and the rate distortion bound.
- To learn the design of entropy codes including Huffman codes and arithmetic coding.
- To understand the operation of lossless compression schemes.
- To understand the operation of popular lossy compression schemes including delta modulation, differential pulse code modulation, transform coding, and vector quantization.

Course Outcome

Students will

- get familiar with concepts of data compression.
- be able to implement lossless and lossy compression schemes

Course Outline

Unit I

Introduction to database systems, file systems Vs DBMS, view of data – data abstraction, view levels, data models, instances and schemas, data independence, database languages, database architecture, database users, database administrator, role of DBA. The entity – relationship (ER) model - entity sets, relationship sets, attributes, constraints, mapping cardinalities, keys, ER diagrams, weak entity sets, strong entity sets.

Unit II

Dictionary methods - string compression, LZ77 sliding window, MZW, GIF images. Image compression - approaches to image compression, intuitive methods and image transform, test images, JPEG, progressive image compression, vector quantization.

Unit III

Wavelet methods - Fourier transform, frequency domain, Fourier image compression, CWT and inverse CWT, Haar transform, filter bank, DWT, JPEG 2000. Video compression - analog video, composite and component video, digital video, video compression, MPEG.

Unit IV

Audio compression - sound, digital audio, human auditory system, MPEG-1 audio layer. Fractal based compression - IFS. Comparison of compression algorithms. Implementation of compression algorithms.

References

1. David Solomon, *Data Compression: The Complete Reference*, 4th Edition, Springer, ISBN: 8184898002.
2. Stephen Welstead, *Fractal and Wavelet Image Compression Techniques*, Lap Lambert Academic Publishing, ISBN: 384651845X.
3. Khalid Sayood, *Introduction to Data compression*, 4th Edition, Elsevier India Pvt. Ltd, ISBN: 8131234088.

MCS4E03b | Pervasive Computing

Course Number: 4.1b
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 75 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To provide a sound conceptual foundation in the area of Pervasive Computing aspects.
□ To provide the students the ability to conceptualize, analyze and design select classes of pervasive computing systems.

Course Outcome

Students will get the ability to conceptualize, analyze and design select classes of pervasive computing systems.

Course Outline

Unit I

Introduction to pervasive computing - past, present, future - the pervasive computing market, m-Business, challenges and future of pervasive computing. Application examples of pervasive computing: retail, airline check-in and booking, sales force automation, healthcare, tracking, car information systems, Email access via WAP and voice.

Unit II

Device technology for pervasive computing - hardware, human-machine interfaces, biometrics, operating systems, Java for pervasive devices, outlook. Device connectivity - protocols, security, device management.

Unit III

Web application concepts for pervasive computing - history, WWW architecture, protocols, trans-coding, client authentication via the Internet for pervasive computing. WAP and beyond - introduction, components of the WAP architecture, WAP infrastructure, WAP security issues, Wireless Markup Language, WAP push, products, i-Mode, outlook.

Unit IV

Web voice technology - basics of speech recognition, voice standards, speech applications, speech and pervasive computing, security personal digital assistants - history, device categories, personal digital assistant operating systems, device characteristics, software components, standards, mobile applications and personal digital assistant browsers. Server side programming (Java) for pervasive computing - Java 2 Enterprise Edition (Overview), servlets, Enterprise Java Beans, Java Server Pages, Extensible Markup Language, Web Services, Model-View-Controller pattern.

Unit V

Pervasive web application architecture - background, scalability & availability - development of pervasive computing web applications, pervasive application architecture - example pervasive application - introduction, user interface overview, architecture, implementation. Access from PCs - smart-card authentication via the Internet, ordering goods. Access via WAP - WAP functionality, implementation - access from personal digital assistants - extending the example application to personal digital assistants, implementation for synchronized devices, implementation for intermittently connected devices, implementation for connected devices - access via voice: extending the example application to voice access, implementation.

References

1. Jochen Burkhardt, Horst Henn, Stefan Hepper, Thomas Schaec and Klaus Rindtorff, *Pervasive Computing: Technology and Architecture of Mobile Internet Applications*, 14th Edition, Pearson Education, ISBN: 8177582801.
2. Stefen Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*, Wiley India Pvt Ltd, ISBN: 8126527331.
3. Guruduth S. Banavar, Norman H. Cohen and Chandra Narayanaswami, *Pervasive Computing: An Application-Based Approach*, Wiley-Blackwell, ISBN: 0471777404.
4. Frank Adelstein, S K S Gupta, GG Richard and L Schwiebert, *Fundamentals of Mobile and Pervasive Computing*, Tata McGraw-Hill, New Delhi, ISBN: 0070603642.
5. Genco and S. Sorce, *Pervasive Systems and Ubiquitous Computing*, 1st Edition, WIT Press, ISBN: 1845644824.
6. Somprakash Bandyopadhyay, Amitava Mukherjee and Debashis Saha, *Networking Infrastructure for Pervasive Computing Enabling Technologies and Systems*, 1st Edition, ISBN: 8184898037.

MCS4E03c | System Security

Course Number: **4.1c**
Contact Hours per Week: **5**
Number of Credits: **3**
Number of Contact Hours: **75 Hrs.**
Prerequisite/Exposure:
Course Evaluation: **20% (Internal) + 80% (External)**

Objectives

- To provide an understanding of the differences between various forms of computer security, where they arise, and appropriate tools to achieve them.

Course Outcome

Students will be able propose appropriate tools to achieve the desired level of security.

Course Outline

Unit I

Notion of different types of securities - information security - computer security - security goals, relation between security, confidentiality, integrity, availability and authorization, vulnerabilities - principles of adequate protection. Notions of operating security, database security, program security, network security. attacks - threats, vulnerabilities and controls. The kind of problems - interception, interruption, modification, fabrication. Computer criminals - amateurs, crackers, career criminals. Methods of defence - control, hardware controls, software controls, effectiveness of controls.

Unit II

Program security - secure programs - fixing faults, unexpected behaviour, types of flaws. Non-malicious program errors - buffer overflows, incomplete mediation. Viruses and other malicious code - kinds of malicious code, how viruses attach, how viruses gain control, prevention, control example - the brain virus, the internet worm, web bugs. Targeted malicious code - trapdoors, Salami attack. Controls against program threats - development controls, peer reviews, hazard analysis.

Unit III

Operating system security - protected objects and methods of protection - memory address protection - fence, relocation, base/bounds registers, tagged architecture, segmentation, paging. Control of access to general objects - directory, access control list. File protection mechanism – basics forms of protection, single permissions. Authentication - authentication basics, password, authentication process challenge - response, biometrics. Trusted operating systems - security policies for operating systems, models of security - requirement of security systems, multilevel security, access security, limitations of security systems. Trusted operating system design - elements, security features, assurance, system flaws and assurance methods.

Unit IV

Database Security - security requirements - integrity of database, confidentiality and availability, reliability and integrity, sensitive data, interface, multilevel database, proposals for multilevel security.

Unit V

Administrating security - security planning - contents of a security planning, team members, commitment to a security plan, business continuity plans. Risk analysis - the nature of risk, steps of risk analysis. Arguments for and against risk analysis, organizational security policies - purpose and goals of organizational security. Audience, characteristics of a good security policy. Nature of security policies - data sensitivity policy, government agency IT security policy. Physical security - natural disaster, human vandals, interception of sensitive information.

References

1. C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 4th Edition, Pearson India, ISBN: 9788131727256.
2. Matt Bishop, *Computer Security: Art & Science*, 1st Edition, Pearson, ISBN: 0201440997.
3. William Stallings, *Cryptography and Network Security: Principles and Practice*, 6th Edition, Pearson India, ISBN: 9332518777.
4. Michael E. Whitman and Herbert J. Mattord, *Principles of Information Security*, 4th Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131516458.

MCS4E03d | Molecular Simulation and Modelling

Course Number: 4.1d
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 75 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To understand application of simulation techniques to study molecular dynamics and derive properties.
□ To learn and apply the statistical approaches and models for phylogenetic analysis and tree reconstruction.
□ To understand the basis and nature of protein-protein interactions.
□ To understand principles of docking simulations. .

Course Outcome

Student will be able to

apply simulation techniques to study molecular dynamics.
learn and apply statistical approaches and models for phylogenetic analysis.
understand and apply docking simulations.

Course Outline

Unit I

Overview of molecular modeling - molecular modeling methods - semi-empirical method and empirical method. Model Type - static, dynamic and probabilistic models. Models of growth and decay.

Unit II

System modeling - concept, principles of mathematical modeling, static physical model, stochastic activities, continuous and discrete simulation. Discrete system simulation - probability concepts in simulation, random number generations and their testing, stochastic variable generation. Model execution - event driven versus time driven.

Unit III

Computational gene mapping - genetic mapping, gene expression, gene prediction methods, gene prediction tools, mutational analysis, introduction to restriction mapping and map assembly, mapping with restriction fragment fingerprints, Lander - Waterman statistics. Software Packages for Phylogenetic Analysis - PHYLogeny Inference Package (Phylip), Phylogenetic Analysis using Parsimony (PAUP) and Phylogenetic Analysis by Maximum Likelihood (PAML). Microarray technology - techniques for microarray data analysis - microarray databases. Scatter Plots,Principal Component Analysis, Cluster Analysis, Applications of Microarray Technology.

Unit IV

Structural Modeling: Protein structure prediction - Prediction of protein secondary structure from the amino acid sequences. Prediction of three dimensional protein structure. Protein structure classification: Two major classification schemes - CATH and SCOP. Protein structure prediction: Steps involved in homology modeling. Protein- Protein Interactions: Prediction methods for Protein- Protein interactions.

Protein - protein interaction Databases. Computer Assisted Drug Design (CADD): Protein based drug design cycle, drug discovery pipeline. Docking Simulations: Rigid docking and Flexible docking.

Unit V

Molecular Visualization: Visualization of protein structure, Methods of studying proteins, Proteomics databases, Protein family databases, PDB file format. Software tools for 3D molecular graphic visualization: Rasmol- basic operations and steps in Rasmol to visualize the molecule, advantages of Rasmol, advantages of Swiss-PdbViewer.

Text Books

1. Stephen Misener and Stephen A. Krawetz, *Bioinformatics: Methods and Protocols*, 1st Edition, Humana Press, ISBN: 1617371564.
2. Geoffrey Gordan, *System Simulation*, 2nd Edition, PHI, ISBN: 9788120301405.
3. Tamar Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide*, 2nd Edition, Springer, ISBN: 1461426502.
4. Narsingh Dev, System *Modelling with Digital Computer*, PHI, ISBN: 0138817898.
5. Andrew Leach, *Molecular Modelling: Principles and Applications*, Prentice Hall. 2nd Edition, ISBN: 81317286092001.
6. Prakash S Lohar, *Bioinformatics*, MJP publishers, Chennai, ISBN: 9788180940668.
7. H-D Holtje, *Molecular Modeling - Basic Principles and Applications*, 3rd Edition, Wiley-VCH, ISBN-13: 9783527315680.
8. Alan Hinchliffe, *Molecular Modelling for Beginners*, 2nd Edition, John Wiley and Sons Ltd, ISBN: 9780470513149.
9. N Cohen, *Guidebook on Molecular Modeling in Drug Design*, 1st Edition, ISBN :9780121782450
10. Masatoshi Nei and Sudhir Kumar, *Molecular Evolution and Phylogenetics*, Oxford University Press, ISBN: 0195135857.

References

1. Asheesh Shanker, Vinay Sharma and Ashok Munjal, *A Textbook of Bioinformatics*, 1st Edition, Rastogi Publications, New Delhi, ISBN: 9788171339174.
2. Des Higgins (Ed), Willie Taylor (Ed), *Bioinformatics: Sequence, Structure and Databanks - A Practical Approach*, 3rd Edition, New Delhi Oxford University Press, ISBN: 0195667530.

MCS4E03e | Fundamentals of Big Data

Course Number: 4.1e
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 75 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To cover the basics of big data.
□ To familiarize with big data technology and tools.

Course Outline

Unit I

Introduction to Big Data – definition & 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*, ISBN: 9781118504222.
2. Eelco Plugge, Peter Membrey and Tim Hawkins, *The Definitive Guide to MongoDB: The NOSQL Database for Cloud and Desktop Computing*, 1st Edition, Apress, ISBN: 9781430230519.
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, ISBN: B006UWBBO6.
4. Garry Turkington, *Hadoop Beginner's Guide*, Packt Publishing Ltd, ISBN: 1849517304.

MCS4E03f | Web Engineering

Course Number: 4.1f
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure: None
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

- To understand the concepts, principles, strategies, and methodologies of web applications development.

Course Outcome

Students will be able to

- identify tools and technologies for developing web applications.
- develop web applications and web services

Course Outline

Unit I

Web Engineering (WE) – introduction – motivation – categories & characteristics of web applications – product related, usage related and development related – evolution of WE.

Unit II

Requirements Engineering (RE) for web applications – introduction – fundamentals – sources of requirements – RE activities – RE specifications in WE - RE principles for web applications – adapting RE methods for web applications development – requirement types, notations, tools.

Unit III

Web application architecture – introduction – fundamentals – definition of architecture – developing and characterising architectures – components of a generic web application architecture – layered architecture – database centric architecture - architecture for web document management – architecture for multimedia data.

Unit IV

Modeling web applications – introduction – modeling specifics in WE – levels – aspects – phases of customizations – modeling requirements – hypertext modeling - hypertext structure modeling concepts – access modeling concepts. Web application design – web design from an evolutionary perspective – information design – software design – merging information design & software design – problems and restrictions in integrated web design – a proposed structural approach – presentation design – presentation of nodes and meshes – device independent development – approaches – interaction design – user interaction – user interface organization – navigation design – deigning a link representation – designing link internals – navigation and orientation – structural dialog for complex activities – interplay with technology and architecture – functional design.

Unit V

Testing web applications – introduction – fundamentals – terminology – quality characteristics – test objectives – test levels – role of tester – test specifics in we – test approaches – conventional, agile - test schemes – three test dimensions – applying the scheme to web applications – test methods and techniques – link testing – browser testing – usability testing – load, stress and continues testing – testing security – test-driven development. Web project development – scope – refining frame work activities – building an WebE team - risk management – making schedule – managing quality, change – project tracking.

References

1. Gerti Kappel, Birgit Proll, Siegried Reich and Werner Retschitzegger, *Web Engineering: The Discipline of Systematic Development of Web Applications*, John Wiley and Sons Ltd, ISBN: 9780470064894.
2. Roger S Pressman and David Lowe, *Web Engineering: A Practitioner's Approach*, 1st Edition, Tata Macgraw Hill Publications, ISBN: 9780073523293.
3. Leon Shklar and Rich Rosen, *Web Application Architecture: Principles, Protocols and Practices*, 2nd Edition, Wiley, ISBN: 047051860X.
4. Guy W Leeky-Thompson, *Just Enough Web Programming with XHTML, PHP, and MySQL*, 1st Edition, Cenagage Learning, ISBN: 159863481X.

5. Anders Moller and Michael Schwartzbach, *An Introduction to XML and Web Technologies*, 1st Edition, Pearson Education, New Delhi, 2009.
6. Chrits Bates, *Web Programming: Building Internet Applications*, 3rd Edition, Wiley India Edition, ISBN: 8126512903

Elective IV MCS4E04 | List of Electives

MCS4E04a |Digital Image Processing

Course Number: 4.2a
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% internal and 80% external

Objectives

- To be familiar with processing of the images, recognition of the pattern and their applications.

Course Outcome

Student will be able to

- explain how digital images are represented and manipulated in a computer.
- familiar with image processing algorithms.

Course Outline

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. Richard E Woods and Rafael C Gonzalez, *Digital Image Processing*, 3rd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131726959.
2. B. Chanda and D.D. Majumder, *Digital Image Processing and Analysis*, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120343255.
3. A.K. Jain, *Fundamentals of Digital Image Processing*, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120309294.
4. W.K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4th Edition, John Wiley, ISBN: 0471767778.
5. Milan Sonka, Vaclav Hlavac and Roger Boyle, *Image Processing Analysis and Machine Vision*, 3rd Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131518833.

MCS4E04b | Introduction to blockchain technology

Course Number: 4.2b
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20 % internal and 80% external

Objectives

□ To study the basics of blockchain technology.

Course Outcome

Student will be able to

- undersand and analyse basic concepts of blockchain technology.

Course Outline

Unit I

Introduction to Blockchain: Blockchain, Distributed Ledger. Introduction to BitCoin: Protocols, Proof of Work, Mining, Blockchain Variants. Blockchain Versions. Features of BlockChain: Block, Blockchain Data Structure, Cryptography: Addresses and Key, Hashing, Merkle root Hash Chain. Data Immutability, Tamper Proof, Embedded Security, Forgery Protection, Data Encryption, Counterfeiting Protection, Double Spend Protection, Fault Tolerance, Miner and Incentives, Peer to Peer Network, Distributed Environment, Distributed Consensus, Distributed Data, Blockchain and Trust, Accounting Model, Auditing, Triple Entry, Anonymity.

Unit II

Blockchain Variants: Public Permissionless, Private Permissionless, Public Permissioned, Private Permissioned. Life cycle of a transaction. Record Keeping Model in Blockchain. Consensus Protocol: Proof of Work(PoW), Proof of Stake(PoS), Practical Byzantine Fault Tolerance(pBFT), Proof of Elapsed Time(PoET), Proof of Authority(PoA).

Unit III

Programmable BlockChain, Ethereum: History, Ethereum vs Bitcoin, Working of Ethereum, EVM: State, Accounts, Ethereum Block, Ether: Gas, Gas Price, Gas Limit, Gas Fee. Transaction Life Cycle: Ethereum Transaction. Ethereum Wallets, Ethereum Client: Geth, Connecting to different networks, Creating Private Chain, Creating Accounts. Geth Console: getBalance, getAccount, deploying contract, mining. Ethereum Simulator: Ganache.

Unit IV

Smart Contract: Comparison with traditional contract, Benefits and Features. Solidity: High Level Constructs: pragma, import, contract, contract. State Variable, Data types: value type, reference type.

Operators, Enumerator, memory, storage, constant. Functions: Visibility, Mutability. Branching Statement: if, if..else, Ternary Operator. Looping Statements: while, do while, for loop. Break statement, Continue Statement.

Unit V

Solc: ABI, Bytecodes. Web3: Features, modules, connecting to Ethereum Network. getAccounts, getBalance, communicating to contract, send method and call method.
Various Blockchain Networks: Hyperledger, Corda, IOTA, . Various DApp: Crypto, Ethlance, Golem, uPort, Medical Chain, BankChain.

References

1. Gavin Wood, Andreas M Antonopoulos, ‘Mastering Ehereum’, O’Reilly Media,Inc, Nov 2018.
2. Melanie Swan, ‘Blockchain’

MCS4E04c | Software Development for Portable Devices

Course Number: 4.1c
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% internal and 80% external

Objectives

- Explain the key differences between development of systems to run on mobile devices and typical personal computing.
- Design effective applications for a mobile device by taking into consideration the underlying hardware-imposed restrictions such as screen size, memory size and processor capability.
- Identify potential security issues and suggest mechanisms to ensure the safety of applications on the mobile device.
- To critically analyze and communicate the differences in architecture and specialized topics such as event handling between applications on the mobile device and non-mobile platforms.

Course Outcome

Students will be able to

- understand unique aspects of mobile applications
- get knowledge about how to work in resource sensitive and resolution variant environment
- design and develop mobile applications using application development framework

Course Outline

Unit I

Introduction to Mobile Web (HTML 5) - Semantic Elements – Structural Elements - Basic formatting tags - heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image. Different attributes like align, color, bgcolor, font face, border, size. Navigation Links using anchor tag - internal, external, mail and image links. Lists - ordered, unordered and definition, table tag, HTML5 form controls - form, input types – color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week, text, password, textarea, button, checkbox, radio button, select box, hidden controls, calendar, date, time, email, url, search. Datalist, keygen, output - Introduction to CSS3.

Unit II

jQuery – introduction - Adding jQuery to web pages – downloading – accessing from CDNs - jQuery syntax - jQuery selectors - event methods - ready(), click(), dblclick(), mouseenter(), mouseleave(), mousedown(), mouseup(), hover(), foucs(), blur() - effects – hide, show, fading, sliding, animation - callback functions – chaining - methods for changing and manipulating HTML elements and attributes - adding new elements/content - append(), prepend(), after(), before() – removing elements - remove(), empty() - manipulating CSS3 - dimensions of elements and browser window – traversing – ancestors, descendants, siblings.

Unit III

Introduction to Android and smart phones, Android architecture & virtual machine, mobile technology terminologies, setting up the environment, setting up emulators, Android fundamentals - activities and applications activity life cycles, activity stacks, activity states. Introduction to manifest, resources & R.java, assets, values – strings.xml - form widgets, views, layouts & drawable resources - XML layouts, linear layouts, relative layouts, table layouts, Android widgets, UI XML specifications events, bundles & intents - explicit intents implicit intents event broadcasting with intents event reception with broadcast receivers, adapters and data binding.

Unit IV

Files, content providers and databases - saving and loading files, SQLite databases - Android database design - exposing access to a data source through a content provider content provider registration native content providers, Android Debug Bridge (adb) tool, Linkify.

Unit V

Adapters and widgets, notifications, custom components threads running on UI thread, Worker thread handlers & runnable asyntask (in detail), playing audio and video, recording audio and video, using the camera to take and process pictures. Networking & location based services - live folders, using sdcards – reading and writing, XML parsing - JSON parsing - including external libraries in applications, Map-based activities, Maps via intent and Map activity GPS, location based services configuration, geocoding, accessing phone services (Call, SMS, MMS), network connectivity services, using Wifi & Bluetooth action bar tabs and custom views on action bars.

References

1. Terry Felke-Morris, Web *Development & Design Foundations with HTML5*, 7th Edition, Addison-Wesley, ISBN: 0133571785.
2. *Html 5 Black Book: Covers CSS3, Javascript, XML, XHTML, Ajax, PHP and Jquery*, Kogent Learning Solutions Inc, ISBN: 9350040956.
3. Kessler, *Programming HTML 5 Applications*, OReilly Media, ISBN: 9350235904.
4. Robin Nixon, *Html5 For Ios And Android: Beginner Guide*, 1st Edition, McGraw-Hill Education India Pvt .Ltd, ISBN: 101259003078.
5. Lauren Darcey and Shane Conder, *Android Wireless Application Development : Android Essentials (Volume 1)*, 3rd Edition, Pearson Education, ISBN: 9332518882.
6. Zigurd Mednieks, Rick Rogers, Lombardo John and Blake Meike, *Android Application Development*, 1st Edition, O'Reilly Meida,
7. Reto Meier, *Professional Android 2 Application Development*, 1st Edition, Wiley India Pvt Ltd, ISBN: 8126525

CSS4E04d | Storage Area Networks

Course Number: 4.1d
Contact Hours per Week: 5
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 100% (Internal)

Objectives

- Understand Storage Area Networks (SAN) characteristics and components.
- Learn about the SAN architecture and management.

- Understand about designing and building SAN.

Course Outcome

Course Outline

Unit I

Basic networking concepts and topologies - OSI reference model, common network devices, network topologies, MAC standards - need for storage networks – storage devices - techniques evolution - benefits of SANs - SAN components and building blocks - fibre channel basics - fibre channel topologies, fibre channel layers, classes of service SAN topologies.

Unit II

SAN fundamentals - SAN operating systems software and hardware types of SAN technology - technology and configuration, high scalability and flexibility standards - storage management challenges - networked storage implementation challenges - storage subsystems for video services.

Unit III

Storage networking architecture storage in storage networking - challenges, cost and performance - Network in storage networking - fibre channel, emerging SAN interconnect technologies - basic software, advanced software, backup software implementation strategies.

Unit IV

Storage network management in-band management out-of-band management - SNMPHTTP - TELNET storage network management issues - storage resource management - storage management, storage, systems and enterprise management integration.

Unit V

Designing and building a SAN - design considerations - business requirements - physical layout, placement, storage, pooling, data availability, connectivity, scalability, migration, manageability, fault tolerance and resilience - prevention of congestion – routability - backup and restoration - SAN security & iSCSI technology - basic security guidelines - implementing SAN security - backup and restoration in iSCSI technology - future of SANS.

References

1. Meeta Gupta, *Storage Area Network Fundamentals*, Cisco Press, ISBN: 158705065X.
2. John R. Vacca, *The Essential Guide to Storage Area Networks*, 1st Edition, Prentice Hall, ISBN: 0130935751.
3. Richard Barker and Paul Massiglia, *Storage Area Network Essentials: A Complete Guide to Understanding and Implementing SANs*, Wiley India Pvt Ltd, ISBN: 8126518588.
4. Tom Clark, *Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs*, 2nd Edition, Addison Wesley Professional, ISBN: 0321136500 .
5. Robert Spalding, *Storage Networks: The Complete Reference*, 1st Edition, Tata McGraw-Hill Education, ISBN: 0070532923.
6. Christopher Poelke and Alex Nikitin, *Storage Area Networks for Dummies*, 2nd Edition, ISBN: 9780470385135.
7. Ulf Troppens, Rainer Erkens, Wolfgang Mueller-Friedt, Rainer Wolafka and Nils Haustein, *Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE*, Wiley India Pvt Ltd, ISBN: 8126518324.

MCS4E04e | Semantic Web

Course Number: 4.2e
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 20% internal and 80% external

Objectives

- To discover the capabilities and limitations of semantic web technology for different applications.

Course Outcome

Students will be able to

- discuss fundamental concepts of semantic web- advantages and limits of semantic web.
- use the ontologies of semantic web in the context of computer science

Course Outline

Unit I

Components – types – ontological commitments – ontological categories – philosophical background – knowledge representation ontologies – toplevel ontologies – linguistic ontologies – domain ontologies – semantic web – need – foundation – layers – architecture.

Unit II

Languages for semantic web and ontologies - web documents in XML – RDF - schema – web resource description using RDF - RDF properties – topic maps and RDF – overview – syntax structure – semantics – pragmatics - traditional ontology languages – LOOM - OKBC – OCML - Flogic Ontology Markup Languages – SHOE – OIL – AML – OIL – OWL.

Unit III

Ontology learning for semantic web - taxonomy for ontology learning – layered approach – phases of ontology learning – importing and processing ontologies and documents – ontology learning algorithms – evaluation.

Unit IV

Ontology management and tools - overview – need for management – development process – target ontology – ontology mapping – skills management system – ontological class – constraints – issues. Evolution – development of tools and tool suites – ontology merge tools – ontology based annotation tools.

Unit V

Applications - web services – semantic web services - security issues – current trends.

References

1. Asuncion Gomez-Perez, Oscar Corcho and Mariano Fernandez-Lopez, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, 1st Edition, Springer, ISBN: 1849968845.
2. Grigoris Antoniou and Frank van Harmelen, *A Semantic Web Primer*, The MIT Press, ISBN: 0262012103.
3. Liyand, *Introduction to the Semantic Web and Semantic Web Services*, Chapman, ISBN: 1584889330.
4. Alexander Maedche, *Ontology Learning for the Semantic Web*, Springer, 2002nd Edition, ISBN: 0792376560.
5. John Davies, Dieter Fensel and Frank Van Harmelen, *Towards the Semantic Web: Ontology – Driven Knowledge Management*, 1st Edition, Wiley, ISBN: 0470848677.

6. Dieter Fensel, Wolfgang Wahlster, Henry Lieberman and James Hendler, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, The MITPress, ISBN: 9780262562126.

MCS4E04f | Advanced Java Programming

Course Number: 4.2f
Contact Hours per Week: 5
Number of Credits: 3
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure: None
Course Evaluation: 20% internal and 80% external

Objectives

□ To learn the advanced features of Java programming language that equip the students to develop web based applications with RDBMS.

Course Outcome

Students will be able to

develop web based java applications.

Course Outline

Unit I

RMI & Servlets - introduction, architecture, defining remote objects, creating stubs and skeletons, serializable classes, accessing remote objects, factory classes, dynamically loaded classes, RMI activation, registering remote objects.

Unit II

Servlets, generic servlet, servlets that access request headers, develop servlets that manipulate response headers, HTTP servlets, forms, HTTP protocols - configuring Tomcat Server, servlet context, servlet context listener, servlet chaining.

Unit III

JNDI & EJB - architecture, context initial context class, objects in a context, binding objects, accessing directory services, attributes and attribute interface modifying directory entities, creating directories entities. EJB roles, architecture, container, implementing a basic EJB object, implementing session beans, implementing entity bean, deploying an enterprise bean object.

Unit IV

Java Server Pages (JSP) - developing JSP pages, technology, syntax using scripting elements, syntax using the courier page directive, create and use JSP error pages, building reusable web presentation, components, JSP technology syntax using the include directive, JSP technology syntax using the jsp:include standard action, developing JSP Pages using custom tags, problem with JSP technology scriptlet code, given an existing custom tag library, develop a JSP page using the library, developing a simple custom tag, structure and execution of a custom tag in a JSP page, tag handler class for a simple empty custom tag, custom tag that includes its body in the contour of the HTTP response, tag library description for a simple, empty custom tag.

Unit V

Hybernate - ORM overview - Hibernate overview, environment, configuration, sessions, persistent class - mapping files - mapping types - examples - O/R mappings - annotations - Hibernate Query Language - Hibernate criteria - queries - Hibernate Native SQL, caching, batch processing, interceptors.

References

1. Jason Hunter and William Crawford, *Java Servlet Programming*, 2nd Edition, O'Reilly Media, ISBN: 0596000405.

2. Karl Moss, *Java Servlets*, McGraw-Hill, ISBN: 0074637398.
3. Barry Burd, *JSP: JavaServer Pages*, IDG Books, ISBN: 0764535358.
4. Prashant Sridharan, *Javabeans Developer's Resource*, ISBN: 0138873089.
5. Chuck Cavaness, *Programming Jakarta Struts*, 2nd Edition, O'Reilly Media, ISBN: 0596006519.
6. Madhusudhan Konda, *Just Hibernate: A Lightweight Introduction to the Hibernate Framework*, Oreilly Meida, ISBN: 9781449334376.

MCS4P01 | Project Work

Course Number: 4.3
Contact Hours per Week: 7 Lecture Hrs and 8 Practical Hrs
Number of Credits: 8
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure:
Course Evaluation: 20% (Internal) + 80% (External)

Objectives

□ To give a practical exposure to the process of software development life cycle.

□ To develop a quality software solution by following the software engineering principles and practices. Students are also encouraged to take up a research oriented work to formulate a research problem and produce results based on its implementation/simulation/experimental analysis.

Course Outcome

Course Outline

Major project work is to be done individually by each student, under the guidance of a faculty member of the concerned department.

Guide 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 a real-life application oriented project work or research and development project. 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 understands 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). The format of the report is included in Appendix A .

Students are also encouraged to present their work in IT fest/conference/workshop/journal with the assistance and guidance of the 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 (Refer Section 7) in the final evaluation.

APPENDIX A – Guidelines for Project Report & Layout

Cover Page & First Page

<<TITLE>>

A PROJECT REPORT

SUBMITTED BY

<<NAME OF THE STUDENT>>

FOR THE AWARD OF THE
DEGREE OF MASTER OF SCIENCE (M.SC.) IN
COMPUTER SCIENCE
(UNIVERSITY OF CALICUT)

<<COLLEGE EMBLEM>>
<<NAME OF THE DEPARTMENT>>

<<NAME OF THE INSTITUTION>>
(AFFILIATED TO THE UNIVERSITY OF CALICUT)
<<ADDRESS>>

MONTH
YEAR

Acknowledgement

ACKNOWLEDGEMENT

I would like to thank ………..

Date: Name of the Student

Page *i*

Declaration by the Student

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person or material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Date: Signature:

Name:

Reg. No.:

Page *ii*

Certificate from Guide & HoD

CERTIFICATE

This is to certify that the project report entitled <<TITLE HERE>> submitted by <<Name of the Student>> (Register Number: << Reg, No>>) to University of Calicut for the award of the degree of **Master of Science** (M.Sc.) in **Computer Science** is a bonafide record of the project work carried out by him/her under my supervision and guidance. The content of the report, in full or parts have not been submitted to any other Institute or University for the award of any other degree or diploma.

Signature
<<Name Project Guide>>
<<Designation>>

Signature
<<Name of the HOD>>
<<Designation>>

Place:

Date:

PROJECT EVALUATION REPORT OF THE EXAMINERS

Certified that the candidate was examined by us in the Project Viva Voce Examination held

on .... and his/her Register Number is ....

Examiners:

1.

2.

Page *iii*

Contents

CONTENTS

Page *iv*

Abstract

ABSTRACT

The abstract is a very brief summary of the report's contents. It should be about half a page long. Somebody unfamiliar with your project should have a good idea of what it's about having read the abstract alone and will know whether it will be of interest to them.

An abstract is a section at the beginning of a report, dissertation, thesis or paper summarising the contents, significant results and conclusions of said document. It allows people to rapidly ascertain the documents purpose and if the document will be useful for them to read.

The abstract is not the same as a summary in the sense you are think of. It is a standalone account of the document giving purpose of the work (objectives), method used, scope of the work, results, conclusions and recommendations.

The abstract, although it comes first logistically, always should be written at the completion of the other chapters of the project report. It needs to be written last because it is the essence of your report, drawing information from all of the other sections of the report. It explains why the experiment was performed and what conclusions were drawn from the results obtained.

A general guideline for an abstract has five sections or areas of focus: why the experiment was conducted; the problem being addressed; what methods were used to solve the problem; the major results obtained; and the overall conclusions from the experiment as a whole.

Do not be misled, however, from this list into thinking that the abstract is a long section. In fact, it should be significantly shorter than all of the others. All of this information should be summarized in a clear but succinct manner if the abstract is going to be successful. An estimated average length for all of this information is only a single paragraph. Although this may seem as though it is a short length to contain all of the required information, it is necessary because it forces you to be accurate and yet compact, two essential qualities.

There are many useful web pages such as http://writing2.richmond.edu/training/proiect/biologv/abslit.html to get few sample abstracts and the common mistakes we make when we write an abstract.

Page *v*

List of Figures

LIST OF FIGURES

Page *vi*

List of Tables

LIST OF FIGURES

Page *vii*

CHAPTER 1

INTRODUCTION

This is a general introduction about the project. Briefly summarize the relevance and background information about the proposed work. It should have the following sections.

1. About the proposed work, underlying technologies and techniques – outline briefly the technological/engineering/scientific/socioeconomic/relevance or significance of the project work being reported.
2. Project Profile – Tile, Area and Category and other relevant information.
3. About the Organization – to whom the Project Work is carried out.
4. Major Contributions of the Project Work.

CHAPTER 2

PROBLEM DEFINITION AND METHODOLOGY

This chapter is meant for giving a detailed description about the problem. This chapter includes the following subsections.

1. Problem Definition
2. Objectives
3. Motivation
4. Methodology
5. Scope

CHAPTER 3

REQUIREMENT ANALYSIS AND SPECIFICATION

This chapter includes the following subsections.

1. Requirement Analysis/Literature Review
2. Existing System
3. Proposed System
4. Requirement Specification
   a. Functional Requirements
   b. Non-functional Requirements
   c. Environmental Details (Hardware & Software Requirements)
5. Feasibility Study
   a. Technical Feasibility
   b. Economical Feasibility
   c. Operational Feasibility
6. Project Planning and Scheduling
   a. PERT Chart
   b. GANTT Chart
7. Software Requirement Specifications (IEEE format preferred)

CHAPTER 4

REQUIREMENT ANALYSIS AND SPECIFICATION

This chapter includes the following subsections.

1. Users of the System
2. Modularity Criteria
3. Architecture Diagrams (whichever of the following if applicable)
   a. DFD
   b. UML Diagrams
   c. Flowchart
4. User Interface Layout
5. Structure of Reports Being Created
6. Database Design
   a. List of Entities and Attributes
   b. E R Diagram
   c. Structure of Tables

CHAPTER 5

IMPLEMENTATION

This chapter is about the realisation of the concepts and ideas developed earlier. It can also describe any problems that may have arisen during implementation and how you dealt with them.

Do not attempt to describe all the code in the system, and do not include large pieces of code in this section. Instead pick out and describe just the pieces of code which, for example:

- Are especially critical to the operation of the system;
- You feel might be of particular interest to the reader for some reason;
- Illustrate a non-standard or innovative way of implementing an algorithm, data structure, etc.

You should also mention any unforeseen problems you encountered when implementing the system and how and to what extent you overcame them. Common problems are:

- Difficulties involving existing software, because of,
  e.g., o its complexity,
  o lack of documentation;
  o lack of suitable supporting software;
  o over-ambitious project aims.

A seemingly disproportionate amount of project time can be taken up in dealing with such problems. The Implementation section gives you the opportunity to show where that time has gone.

Complete source code should be provided separately as an appendix.

This chapter includes the following subsections.

1. Brief description about the Tools/Scripts for Implementation
2. Module Hierarchy
3. Coding
4. Problems Encountered

CHAPTER 6

TESTING

This chapter includes the following subsections.

1. Test Plans
2. Unit Testing
    a. Test Items (Test Cases)
3. Integration Testing
4. System Testing
    a. Test Items (Test Cases)
5. Implementation - Changeover Plans

CHAPTER 7

CONCLUSION

The purpose of this section is to provide a summary of the whole thesis or report. In this context, it is similar to the Abstract, except that the Abstract puts roughly equal weight on all report chapters, whereas the Conclusion chapter focuses primarily on the findings, conclusions and/or recommendations of the project.

There are a couple of rules for this chapter:

- All material presented in this chapter must have appeared already in the report; no new material can be introduced in this chapter (rigid rule of technical writing).
- Usually, you would not present any figures or tables in this chapter (rule of thumb).

Conclusions section can have the following (typical) content. These contents must **not** be given in bulleted format.

- Re-introduce the project and the need for the work though more briefly than in the introduction.
- Reiterate the purpose and specific objectives of your project.
- Recap the approach taken similar to the road map in the introduction.
- However, in this case, you are re-capping the data, methodology and results as you go.
- Summarize the major findings and recommendations of your work.

Future Enhancements

Identify further works that can be added to make your system to meet the challenges of tomorrow. You can also include whatever requirements you could not fully due to the scarcity of time/resources.