FAROOK COLLEGE (AUTONOMOUS)

Farook College PO, Kozhikode-673632

P.G Programme in Computer Science

Under

Choice Based Credit Semester System

SYLLABUS

(2022 Admission Onwards)

Prepared By:

Board of Studies in Computer Science

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 2022 admission onwards.

Date: Principal
Place: Farook College

M.Sc. Computer Science Programme:

The M.Sc. Computer Science Programme of the department of Computer Science at Farook College (Autonomous) , started in 2001, aims to provide the students with state of the art knowledge, general competence, and technical skills on an advanced level needed in academics, industry and research. The students are able to acquire the ability to design solutions or systems for complex problems according to the specifications given. The programme also provides students an orientation to pursue their career in Research and Development.

Programme educational objectives(PEOs):

The following Programme Educational Objectives are designed based on the department mission.

The PEO's are to facilitate post-graduating students to

PEO 1: acquire advanced knowledge and expertise necessary for professional practice in Computer Science and its applications for employment, higher studies or research.

PEO 2: attain and practice technical skills to identify, analyze and solve complex problems and issues related to Computer Science.

PEO 3: possess a professional attitude as an individual or a team member with consideration for society, professional ethics, environmental factors and motivation for life-long learning.

Programme Outcomes (POs):

The M.Sc. Computer Science Programme develops human resource for government organizations, R & D institutions and IT industries. It also equips the students to start their own business or to become professionals as a software developer, database administrator, programmer, system analyst, data scientist, web application developer, system programmer, software testing and expert system designer.

Knowledge outcomes:

Students will be able to

PO1: Acquire the skills to find appropriate and effective solutions needed for the situations by applying knowledge gained through different programming languages, tools and software covered in the syllabus of programme.

PO2: To get working experience within a specific field of Computer Science, through project work.

PO3: To get the ability to apply knowledge of Computer Science to the real-world issues.

PO4: Be familiar with current research issues and problems within various fields of Computer Science.

PO5: Use creativity, critical thinking, analysis and research skill.

Skill outcomes:

Students will be able to

PO6: Learn latest technologies, embracing the concepts and issues in its use, especially in the field of computers.

PO7: Grab placements by developing personality, soft skills & Technical skills.

PO8: Communicate scientific information in a clear and concise manner.

PO9: Build up programming, analytical and logical thinking abilities.

General Competency Outcomes:

Students will be able to

PO10: Be able to understand the role of Computer Science in solving present-day problems in society.

PO11: Know the recent developments in IT, future scope and limitations, and catch on the value of lifelong learning.

PO12: Get an ability to participate in debates, discussions in the society constructively.

Programme Specific Outcomes (PSOs):

After completing M.Sc. Computer Science Program students will be able to:

PSO1: Enrich the knowledge in the areas like Artificial Intelligence, Big Data, Data mining, Data Analysis, Cloud Computing, Paradigm of Programming language, Design and Analysis of Algorithms, Database Technologies Advanced Operating System, Mobile Technologies, Software Engineering and core computing subjects.

PSO2: Students understand all dimensions of the concepts of software application and projects.

PSO3: Demonstrate the theoretical concepts with the use of ICT.

PSO4: Develop in-house applications as terms of projects.

PSO5: Interact with IT experts enrich knowledge.

PSO6: Get industrial exposure through the Industrial Internship and IT industry visits.

PSO7: Get opportunity to enrich knowledge by becoming the part of professional & tinkering communities.

PSO8: To make them employable according to current demand of IT Industry and responsible citizen.

PSO9: Get trained to publish their work in journals/conferences/IT fest through the knowledge gathered from term papers.

M.Sc. Computer Science Curriculum Structure & Syllabus
(2022 - Admission Onwards)
Total Credits = 80

Semester I

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**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	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 and Architecture	4	1	4	5	4	0	4
1.6	MCS1L01	Practical 1	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 (Excluding Audit Course)			**22**				**20**	**5**	**25**

Semester II

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**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	Data Analysis & Visualisation using Python	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 (Excluding Audit Course)			**22**				**20**	**5**	**25**

Table Legend[1]

[1]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

Semester III

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**I**	**E**	**T**	**L**	**P**	**T**
3.1	MCS3C01	Advanced Database Management System	4	1	4	5	4	1	5
3.2	MCS3C02	Fundamentals of Artificial Intelligence and Machine Learning	4	1	4	5	4	0	4
3.3	MCS3C03	Object oriented Programming Concepts	4	1	4	5	4	0	4
3.4	MCS3E01	Elective 1	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	**22**				**20**	**5**	**25**

Elective I - MCS3E01 : List of Courses

Course		Title
No.	**Code**	
E1.1	MCS3E01a	Computer Graphics
E1.2	MCS3E01b	Introduction to Soft Computing
E1.3	MCS3E01c	Principles of Compilers
E1.4	MCS3E01d	Bioinformatics
E1.5	MCS3E01e	Computer Optimization Techniques
E1.6	MCS3E01f	Numerical and Statistical Methods

Elective II - MCS3E02 : List of Courses

Course		Title
No.	**Code**	
E2..1	MCS3E02a	Pattern Recognition
E2.2	MCS3E02b	Wireless and Mobile Networks
E2.3	MCS3E02c	Cryptography and Network Security
E2.4	MCS3E02d	Advanced Web Technology
E2.5	MCS3E02e	Virtualization and Cloud Computing
E2.6	MCS3E02f	Data Warehousing and Data Mining

Semester IV

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**I**	**E**	**T**	**L**	**P**	**T**
4.1	MCS4E03	Elective III	3	1	4	5	5	0	5
4.2	MCS4E04	Elective IV	3	1	4	5	5	0	5
4.3	MCS4P01	Project Requirement Analysis and Design – Related Discussion	8	1	4	5	3	1	4
		Project Coding, Testing & Implementation - Related Discussion					2	2	4
		Project Evaluation and Assessment					2	0	2
		Project Lab Work					0	5	5
		Total	**14**				**17**	**8**	**25**

Elective III- MCS4E03 : List of Courses

Course		Title
No.	**Code**	
E3.1	MCS4E03a	Data Compression
E3.2	MCS4E03b	Pervasive Computing
E3.3	MCS4E03c	System Security
E3.4	MCS4E03d	Molecular Simulation and Modelling
E3.5	MCS4E03e	Fundamentals of Big Data
E3.6	MCS4E03f	Web Engineering

Elective IV- MCS4E04 : List of Courses

Course		Title
No.	**Code**	
E4.1	MCS4E04a	Digital Image Processing
E4.2	MCS4E04b	Introduction to Block chain technology
E4.3	MCS4E04c	Software Development for Portable Devices
E4.4	MCS4E04d	Storage Area Networks
E4.5	MCS4E04e	Semantic Web
E4.6	MCS4E04f	Advanced Java Programming

First Semester

Semester I

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**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	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 and Architecture	4	1	4	5	4	0	4
1.6	MCS1L01	Practical 1	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 (Excluding Audit Course)			**22**				**20**	**5**	**25**

[2]Table Legend

[2]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS1C01:Discrete Mathematical Structures

Course Number: 1.1
Hours per Week: 4
Credits: 4

Course Description:

This course helps the learner to apply the theory and applications of elementary Set Theory, Functions and Relations, Propositional Logic, Predicate Logic, Lattices, Algebraic Structures and Graph theory eventually in practical applications.

Course Outcomes:

C01 Interpret the fundamental ideas of set theory.

C02: Check the validity of predicates in Propositional and Quantified Propositional Logic using truth tables, deductive reasoning and inference theory on Propositional Logic

C03: Classify different types of functions and identify their properties. Classify binary relations into various types and illustrate an application for each type of binary relation, in Computer Science

C04: Solve counting problems by applying the elementary counting techniques Pigeonhole Principle and Principle of Inclusion and Exclusion

C05: Illustrate an application for Partially Ordered Sets and Complete Lattices, in Computer Science

Prerequisites:

- Background of the basic mathematics at +2 level.

Course Outline:

Module 1: Sets and Mathematical Logic: Set Theory - Types of sets, Set operations, Principles of Inclusion and Exclusion. Mathematical Logic - Propositional Calculus - Statement, Connectives, Conditional and bi-conditional, 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: 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.

References:

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 Pblications 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. Garnier, John Taylor, Discrete Mathematics Proofs, Structures and Applications.IoP
5. Kenneth H .Rosen, "Discrete Mathematics and its Applications", 5/e, Tata Mc Graw Hill Pub. Co. Ltd, New Delhi 2003

MCS1C02:Advanced Data Structures

Course Number: 1.2
Hours per Week: 4
Credits: 4

Course Description:

This course introduces basic and advanced data structures that are essential for the development of programs. Also it focuses on how to effectively use various data structures for the development of programs for problem solving.

Course Outcomes:

CO1: Students will be able to demonstrate the concepts of data structures and running time of algorithms/programs.

CO2: Students will be able to analyse algorithms.

CO3: Students will be able to demonstrate the basic data structures.

CO4: Students will be able to demonstrate advanced data structures.

CO5: Students will be able to implement basic and advanced data structures for problem solving.

Prerequisites:

- Understanding of programming languages.

Course Outline:

Module 1: 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.

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.

Searching and Sorting - sorting algorithms - linear search & binary search – complexity.

Module 2: 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

Module 3: Non-linear Data Structures : Trees–terminology-tree traversals algorithms - Binary trees - threaded binary trees – binary search trees - traversals and operations on BST. Balanced trees - m-way trees, B and B^+ trees.

Graphs - representation of graphs - operations - traversals and their implementation.

Module 4: Advanced Tree Structures & Heap : Red Black Tree, Splay tree, Digital Search Tree, Tries, Treaps, Huffman algorithm for extended binary tree – operations and their implementation.
Heap structures - overview of heaps - implementation and operations, Min-Max heaps - Deaps - leftist heaps - binomial heaps -binary heaps - skew heaps – applications - amortized analysis-an unrelated puzzle - Binomial queues.

Module 5: 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.

References:

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

3. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of DataStructures in C, Silicon Press, ISBN: 0929306406.
4. Richard F. Gilberg and Behrouz A. Forouzan, Data Structures: A PseudocodeApproach With C, Thomson Brooks/Cole Publications, Course Technology, ISBN:9780534390808.
5. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, DataStructure using C, Prentice- Hall, ISBN: 9780131997462.
6. Robert Kruse, Tondo C L and Bruce Leung, Data Structures Program Design in C, Pearson India, 2ndEdition, 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 DataStructures with Applications, 2ndEdition, Mcgraw-Hill College, ISBN: 0070651574.

MCS1C03:Theory of Computation

Course Number: 1.3
Hours per Week: 4
Credits: 4

Course Description:

This is a core course in theoretical computer science. It covers automata and grammar representations for each and every language in Chomsky Hierarchy. The topics covered in this course have applications in various domains including compiler design, decidability and complexity theory, software testing, formal modelling and verification of hardware and software.

Course Outcomes:

CO1: Students will be able to describe the concept of automata and be able to design DFA, NFA and NFA with epsilon moves for a given problem.

CO2: Students will be able to find out the regular expression for a regular language and be able to design automata for a given regular expression.

CO3: Students will be able to use pumping lemma and find out whether the given language is regular or not.

CO4: Students will be able to describe context free languages in Chomsky Normal Form and Greibach Normal Form.

CO5: Students will be able to use pumping lemma and find out whether the given language is context free or not.

CO6: Students will be able to design push down automata – the machine part of context free language- for a given CFL.

CO7: Students will be able to describe Turing machines Working and be able to design a Turing Machine for a given problem.

CO8: Students will be able to describe Chomsky Hierarchy and also be able to classify any given language into its corresponding class of language or grammar.

CO9: Students will be able to describe the concepts computability and decidability and also will be able to describe the complexity classes.

Prerequisites:

- Understanding of any of the programming languages.

Course Outline:

Module 1: 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.

Module 2: Regular Expressions, Finite Automata and Regular Expressions, Properties of Regular Languages, Pumping lemma and proof for existence of nonregular languages, Closure properties, homomorphism, substitution, Myhill Nerode Theorem and DFA state minimization, Regular Grammar.

Module 3: Context Free Languages - Equivalence of CFG and PDA (Proof not needed) , 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 noncontext free languages, Context Sensitive languages, Equivalence of LBA and Context Sensitive Grammar (CSG).

Module 4: 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.

Module 5: 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 AutomataTheory, Languages of Computation, 3rdEdition, 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, Available at https://cs.brown.edu/people/jsavage/book/pdfs/ModelsOfComputation.pdf
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
Hours per Week: 4
Credits: 4

Course Description:

The purpose of this course is to learn the fundamentals of problem solving with the help of algorithms and flowcharts. The course helps the learner to develop correct solutions for a problem by making use of the fundamental design constructs - sequence, selection and iteration. Learners are also exposed to the technique of creating modules in the solutions. Learners can also implement the solutions using the C programming language.

Course Outcomes:

CO1: Students will be able to identify the fundamental principles of developing solutions using flowcharts and algorithms.

CO2: Students will be able to design correct solutions using algorithms and flowcharts making use of sequence, selection and iteration logic constructs

CO3: Students will be able to develop solutions for problems following a systematic approach.

CO4: Students will be able to design solutions using modular programming approach

CO5: Students will be able to design solutions using iteration and recursion.

CO6: Students will be able to implement solutions in C programming language.

Prerequisites:

- Basic knowledge in mathematics.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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).

Module 4: 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.

Module 5: 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, Laxm 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
Hours per Week: 4
Credits: 4

Course Description:

This course is intended to familiarize with the digital fundamentals, computer organization, computer architecture and microprocessors.

Course Outcomes:

CO1: Learners will be able to explain the working of different combinational and sequential circuits.

CO2: Learners will be able to

- Explain the basic computer organization.
- Explain different stages of the Instruction Cycle.
- Distinguish working of different Instructions.
- Analyse the effect of different addressing modes.
- Explain the working of different types of control units.

CO3: Learners will be able to explain and summarize the implementations of various arithmetic operations in ALU.

CO4: Learners will be able to

- Differentiate various memory elements
- Illustrate the working of Cache memory and Virtual memory
- Explain Interrupts and their handling
- Differentiate different ways to access I/O devices

CO5: Learners will be able to explain the architecture and working of various microprocessors like 8085 and 8086.

Prerequisites:

- Knowledge in Boolean Algebra & basic electronics.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: Arithmetic & Logic Unit - addition of positive numbers – signed addition and subtraction - addition/subtraction logic unit – multiplication of positive numbers – signed number multiplication - multiplication using Booth's algorithm - division-restoring and non-restoring algorithms, floating point numbers and operations.

Module 4: Main Memory - memory hierarchy – main memory – RAM, ROM - Cache memory – Address Mapping – Direct mapping- Associate mapping – Set associate mapping - memory management requirements - secondary storage – Hard disk - memory interleaving. Input / Output Organization - Accessing I/O devices – programmed I/O, interrupt I/O - interrupts - interrupt processing – Daisy chaining - direct memory access (DMA) - DMA operations DMA Controller, Introduction to I/O interfaces, I/O channels, IO Processors.

Module 5: 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
Hours per Week: 4
Credits: 2

PART A - Art of Programming Methodology

The purpose of this course is to provide hands-on training for programming using C Language and to up-skill the problem solving capability of the learner. Learners are exposed to developing solutions using the basic constructs of programming. Learners are expected to use laboratory and practice the problems listed in the programming cycle of C.

Course Outcomes:

CO1: Students are able to design and develop programs using C Language.

CO2: Students are able to use appropriate data structures for programming.

CO3: Students are able design solutions to scientific or mathematical problems.

Course Outline:

1. Simple C Programs like the 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. Pointers - simple programs to learn the concept of pointers, array operation using pointers etc.
6. Writing functions.

7. Implementation of recursion (recursive function to compute a factorial, reverse string etc).
   SEE Appendix A.1 for detailed Program List

PART B - Advanced Data Structures:

The purpose of this course is to use laboratory and practice implementing programs listed here.

- Experiments are to be done using C Programming language.
- Lab cycle for **PART B** consists of 25 experiments.
- Minimum number of experiments in the Laboratory Record Book should be 20.
- Laboratory Record Book for **PART B** should include Experiment No, Date, Aim of experiment, Algorithm, output and the result.

Course Outcomes:

CO1: Students are able to implement various data structures.

CO2: Students are able to use appropriate data structures for programming.

CO3: Students are able to apply data structures for solving problems effectively.

Course Outline:

1. Implementation of Stack using Array.
2. Implementation of Queue using Array.
3. Implementation of Circular Queue using Array.
4. Apply stack to convert decimal number to binary.
5. Implementation of Sequential Search.
6. Implementation of Binary Search.
7. Implementation of Linked list operations
   (a) Add,

(b) Insert,

(c) Delete, and

(d) Search

8. Implementation of Stack using Linked List.
9. Implementation of Queues using Linked List.
10. Implementation of Circular Linked List.
11. Implementation of Binary Tree and its traversal.
12. Conversion of an Infix expression to Postfix using Stack.
13. Evaluation of a Postfix expression.
14. Implementation of Conversion of an Infix expression to Prefix using Stack.
15. Implementation of Heap Tree and operations.
16. Implementation of Insertion Sort.
17. Implementation of Merge Sort.
18. Implementation of Quick Sort.
19. Implementation of Heap Sort.
20. Representation Sparse Matrix using Array and perform,

    (a) Matrix Addition,

    (b) Simple Transpose and

    (c) Fast Transpose.

21. Implementation of Doubly Linked List.
22. Representation of a graph using adjacency matrix.
23. Implementation of Graph Searching (DFS & BFS).
24. Implementation of Prims algorithm.
25. Implementation of Dijkstra's shortest path algorithm.

MCS1A01: Introduction to Research (Ability Enhancement Audit Course)

Course Number: 1.7
Number of Credits: 4

Course Description:

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:

- Identify the components of a literature review process
- Critically analyse 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 examinations 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 enrolled 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 Outcomes:

CO1: Students are able to describe research terminology.

CO2: Student will be able explain ethical principles of research.

CO3: Identify the components of a literature review process.

CO4: Critically analyse published research.

CO5: Students will be able to explain research methods in the field of computer Science.

Prerequisites:

- Basic knowledge in using computer and ability to understand English.

Course Outline:

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

Week 1: Overview of Research

Week 2: 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

Week 3: Data Analysis

Week 4: How to make Technical presentation – Technical Writing

Week 5: Intellectual property

Week 6: Research in Computer Science Engineering

References:

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

Second Semester

Semester II

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**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	Data Analysis& Visualisation using Python	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 (Excluding Audit Course)			**22**				**20**	**5**	**25**

[3]Table Legend

[3]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS2C01:Design and Analysis of Algorithms

Course Number: 2.1
Hours per Week: 4
Credits: 4

Course Description:

The course introduces students to the design of computer algorithms, as well as analysis of algorithms. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms.

Course Outcomes:

CO1: Identify the suitable design strategy to solve a given problem.

CO2: Analyse any given algorithm and express its time and space complexities in asymptotic notations.

CO3: Solve recurrence equations using Iteration, Recurrence Tree, Substitution and Master's Method to compute time complexity of algorithms.

CO4: Demonstrate Divide-and-conquer, Greedy Strategy, Dynamic programming, Branch-and Bound and Backtracking algorithm design techniques

CO5: Classify a problem as computationally tractable or intractable, and discuss strategies to address intractability.

CO6: Identify parallel processing algorithms and the associated properties.

Prerequisites:

- Basic knowledge in Data structures

Course Outline:

Module 1: 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 2: 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. Analysing 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 3: 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 4: 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 5: Analysing 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)

MCS2C02:Operating System Concepts

Course Number: 2.2
Hours per Week: 4
Credits: 4

Course Description:

The purpose of this course is to learn the underlying principles of an operating system. The learner is exposed to the topics such as principle of concurrency, memory management, uni-processor management and multiprocessor management. The course concludes with a fundamental understanding of client server computing and Service-Oriented Architecture.

Course Outcomes:

CO1: Students will be able to identify the fundamental principles & functions of operating systems.

CO2: Students will be able to explain the fundamental principles of concurrency mechanism in Operating Systems.

CO3: Students will be able to demonstrate the various resource management strategies & techniques in Operating Systems.

CO4: Students will be able to identify the techniques for achieving security in Operating Systems.

CO5: Students will be able to identify the fundamental concepts of Operating System of IOT.

Prerequisites:

- Knowledge in Data structures.

Course Outline:

Module 1: 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

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: Security concepts: CIA Triad - Techniques to Achieve Security - Authentication - Access Control - Intrusion Detection - Malware Defense - Buffer Overflow Attacks - Access Control - Hardening - Security Maintenance. Case Study: Access Control in Unix and Windows.

IoT Operating Systems - Constrained Devices - Classification - Requirements for an IOT OS - Architecture of IoT OS - Case Study: RIOT.

References:

1. William Stallings, Operating Systems, Internals and Design Principles, 9th Edition, Pearson, ASIN : 1292214295

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
5. Distributed, Database, and Multiprocessor Operating Systems, Tata McGraw-Hill Education Private Limited, ISBN: 9780070575721.

MCS2C03:Computer Networks

Course Number: 2.3
Hours per Week: 4
Credits: 4

Course Description:

Study of this course provides the learners a clear understanding of how computer networks from local area networks to the massive and global Internet are built, and how they allow computers to share information and communicate with one another.

Course Outcomes:

CO1: Students will be able to describe how the internet works and also be able to compare ISO OSI reference model and TCP IP reference model.

CO2: Students will be able to describe the TCP application layer protocols – HTTP, FTP, SMTP, and DNS.

CO3: Students will be able to describe transport layer services and also be able to compare transport layer protocols TCP and UDP.

CO4: Students will be able to describe network layer services and also be able to compare transport layer protocols IP v4 and IP v6.

CO5: Students will be able to describe link layer services and also be able to explain link layer protocols like ARP.

CO6: Students will be able to describe the concepts of cryptography and firewalls.

Prerequisites:

- Knowledge in Data structures & Programming.

Course Outline:

Module 1: 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.

Module 2: Application layer protocols – principles – the web and HTTP – FTP – Email in Internet – DNS. Socket programming – building a Web server - content distribution.

Module 3: 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.

Module 4: Link layer services - error detection and correction - multiple access protocols – LAN address – ARP – Ethernet – hubs – bridges – switches - wireless links – PPP - ATM.

Module 5: Security in Networks – Principles of Cryptography – Authentication – Integrity – Key Distribution and Certification – Firewalls – Attacks and Countermeasures.

References:

1. J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach Featuring Internet, 6thEdition, Pearson Education.
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.

MCS2C04: Data Analysis & Visualisation using Python

Course Number: 2.4
Hours per Week: 4
Credits: 4

Course Description:

This course provides a foundation for data analysis using python. Basics of python programming, numpy, pandas and matlibplot are provided in this course.

Course Outcomes:

CO1: Students will be able use python to develop solutions to simple scientific and mathematical problems.

CO2: students will be able do modular programming in python.

CO3: Students will be able to use numpy for array manipulation..

CO4: Students will be able to use pandas for data manipulation.

CO5: Students will be able to use matlibplot to visualize the data.

Prerequisites:

- Basic knowledge in mathematics.

Course Outline:

Module 1: Python Fundamentals for Data Analysis-Python data structures, Control statements, Functions, Object Oriented programming concepts using classes, objects and methods, Exception handling, Implementation of user-defined Modules and Package, File handling in python.

Module 2: Introduction to Data Understanding and Preprocessing Knowledge domains of Data Analysis, Understanding structured and unstructured data, Data Analysis process, Dataset generation, Importing Dataset: Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing the Data: Identify and Handle Missing Values.

Module 3: Numpy - Introduction to numpy, Creating arrays, Using arrays and Scalars, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input and Output.

Module 4: Pandas: Features and uses of pandas - Series in pandas, Index objects, Reindex, Drop Entry, Selecting Entries, Data Alignment, Rank and Sort, Summary Statics, Missing Data, Index Hierarchy. Data Manipulation with Pandas-manipulate textual data, Pandas Objects -series and data frame, Creating Series from simple datatypes, Load CSV files to Python Pandas

Module 5: Python For Data Visualisation: Data Visualization, Families of Visualizations- charts, tables, Plotting and Visualization, Graph Plotting in Python using Matplotlib - Line Plot, Bar Plot, Scatter Plot, pie chart , Three-Dimensional Plotting in Matplotlib, Visualizing with NumPy, Create plots using NumPy Array, plot using Pandas.

References:

1. 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/
2. Richard L. Halterman, Learning To Program With Python
3. Python Documentation. Available at http://www.python.org /doc/ Swarooph CH, A Byte of Python. Available at http://swaroopch.com /notes/ python/
4. Wesley J Chun, Core Python Programming, 2nd Edition, Pearson Education, ISBN: 8131711889.
5. Hands-On Data Analysis And Visualization With Pandas, Book by PURNA CHANDER RAO. KATHULA, 2020.
6. Thinking in Pandas: How to Use the Python Data Analysis Library the Right Way, Books By Hannah Stepanek, 2020.
7. Hands-On Data Analysis And Visualization With Pandas, Books By Daniel Y. Chen, April 2018.
8. Pandas Tutorial : https://pandas.pydata.org /pandas-docs /stable/ getting_started /tutorials.html.
9. The Python Tutorial, https://docs.python.org/3/tutorial
10. NumPy Quick Start, https://numpy.org/doc/stable /user/quickstart.html

MCS2C05:Principles of Software Engineering

Course Number: 2.5
Hours per Week: 4
Credits: 4

Course Description:

The course includes important concepts of software engineering including software process models, requirement analysis, planning, design, coding, testing and implementation.

Course Outcomes:

CO1: Students will be able to summarize about the fundamental concepts software engineering and various software process models

CO2: Students will be able to explain the process of software requirement analysis, the structure, components and characteristics of SRS

CO3: Students will be able to outline the activities in software planning. They will be able to classify various software risks and formulate risk management activities.

CO4: Students will be able to explain the software design process. They will be able to compare structured and function-oriented design techniques and tools

CO5: Students will be able to summarize the coding and testing process. They will be able to summarize various testing approaches

Prerequisites:

- Knowledge in Programming language and basic mathematics.

Course Outline:

Module 1: 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 2: Software Requirements Analysis and Specification –Software Requirements –Problem Analysis –Requirements Specification –Validation –Metrics.

Module 3: 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 4: 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 5: 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
Hours per Week: 4
Credits: 4

PART A - Operating System

The purpose of this course is to get a practical exposure to the underlying principles of an operating system. The learner can develop algorithms and prepare programs in C programming language to understand the fundamentals of process creation, blocking and its execution. Learner is also exposed to hands-on experience on the demonstration of algorithms for uniprocessor management, concurrency mechanism using semaphores and memory management schemes.

Course Outcomes:

CO1: Students will be able to explain the use of system calls in operating systems.

CO2: Students will be able to implement creation, blocking and execution of processes using relevant system calls.

CO3: Students will be able to implement interprocess communication using relevant system calls.

CO4: Students will be able to demonstrate process synchronization by implementing producer consumer problem using semaphores.

CO5: Students will be able to implement memory allocation schemes.

CO6: Students will be able to implement CPU scheduling algorithms.

CO7: Students will be able to implement file management techniques (reading, writing, appending).

Course Outline

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 display 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 display 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 display 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.

PART B - Data Analysis with Python

The purpose of this course is to get hands on experience to the students in using python for data analysis and visualisation. The learner can develop programs using various python programming constructs and data types. The focus is given to numpy and pandas.

CO1: Students will be able use python to develop solutions to simple scientific and mathematical problems.

CO2: students will be able practice modular programming in python.

CO3: Students will be able to use numpy for array manipulation..

CO4: Students will be able to use pandas for data manipulation.

CO5: Students will be able to use matlibplot to visualize the data.

Course Outline:

1. Python programs using Loops & control statements.
2. Python programs using functions.
3. Python programs for file handling.
4. Basic programs using numpy.
5. Python programs for array manipulation.
6. Installing & loading pandas
7. Programs using pandas.
8. File and Data manipulation using pandas.
9. Installing matplotlib in python.
10. Programs for data visualisation (line,Bar, Pi, scatter plot/charts) using matlibplot.
11. **SEE Appendix A.2** for detailed Program list.

MCS2A02:Term Paper (Professional Competency Audit Course)

Course Number: 2.7
Credits: 4

Course Outcomes:

CO1: Students are able to conduct literature survey in any specific research area

CO2: Students will be able to use LaTeX and format a document using LaTeX

CO3: Students can identify the steps in writing a research paper

Course Description:

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:

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

Third Semester

Semester III

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**I**	**E**	**T**	**L**	**P**	**T**
3.1	MCS3C01	Advanced Database Management System	4	1	4	5	4	1	5
3.2	MCS3C02	Fundamentals of Artificial Intelligence and Machine Learning	4	1	4	5	4	0	4
3.3	MCS3C03	Object Oriented Programming Concepts	4	1	4	5	4	0	4
3.4	MCS3E01	Elective 1	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	**22**				**20**	**5**	**25**

[4]Table Legend

[4]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS3C01:Advanced Database Management System

Course Number: 3.1
Hours per Week: 4
Credits: 4

Course Description:

This course begins with principles of Database Management Systems (DBMS) and covers advanced databases like object-oriented databases and distributed databases. Also gives an overview of unstructured database NoSQL.

Course Outcomes:

CO1: Students will be able to summarize basic concepts and different models in the database management system.

CO2: Students will be able to design ER diagrams for given real word scenarios described as informal descriptions.

CO3: Students will be able to model and design normalized database solutions for efficiently representing and querying data using relational model.

CO4: Students will be able to describe transaction management concepts and compare the aspects of Concurrency Control and Recovery in Database systems

CO5: Students will be able to describe the object-oriented database concepts, its pros and cons.

CO6: Students will be able to describe the distributed database concepts, concurrency and recovery in distributed database systems.

Prerequisites:

- Knowledge in Data structures, Basic mathematics & Set theory.

Course Outline:

Module 1: Introduction to Database Management Systems (DBMS) - Characteristics of Database system, Database Users, Data Models and Schema - Three Schema architecture. Database Languages, Database architectures and classification.

ER model - Basic concepts, entity set attributes, notations, Relationships and constraints, cardinality, participation, notations, weak entities.

Structure of Relational Databases - Integrity Constraints, Synthesizing ER diagram to relational schema. Relational algebra and calculus.

Module 2: 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, Stored Procedures and triggers.

Module 3: 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.

Module 4: Object Oriented Database Management Systems (OODBMS) - concepts, need for OODBMS, composite objects, issues in OODBMSs, advantages and disadvantages of OODBMS.

Module 5: 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.

References:

1. Elmasri and Navathe, Fundamentals of Database Systems, 5th Edition, Pearson.
2. Abraham Silbersehatz, Henry F. Korth and Sudarshan, Database System-Concepts, 6thEdition, Tata McGraw-Hill.

3. CJ Date, An Introduction to Database Systems, 8th Edition, Addison Wesley.
4. Ramakrishnan and Gehrke, Database Management Systems, 3rd Edition, McGraw - Hill Education.
5. Alexis Leon and Mathews Leon, Database Management Systems, 1st Edition, Vikas Publishers.
6. Vikram Vaswani, MySQL The complete Reference, 1st Edition, Tata Mcgraw Hill Education Private Limited.
7. Joel Murach, Murach's Mysql, Mike Murach Associates Inc.
8. Guy Harrison, Next Generation Data Bases – NoSQL, NewSQL and Big Data, 1stEd ,Apress, 2015.
9. NoSQL Data Models: Trends and Challenges (Computer Engineering: Databases and Big Data), Wiley, 2018

MCS3C02:Fundamentals of Artificial Intelligence and Machine Learning.

Course Number: 3.2
Hours per Week: 4
Credits: 4

Course Description:

This course introduces concepts of artificial intelligence and machine learning that can be applied for the development of modern programs. Also it focuses on how to effectively use various AI & Machine learning techniques and fundamental knowledge in AI so that they can be applied for solving real world problems.

Course Outcomes:

CO1: Students will be able to demonstrate the concepts, characteristics of AI problems and their representations for designing solutions to AI problems.

CO2: Students will be able to demonstrate and apply various searching techniques that applied for finding solutions to AI problems.

CO3: Students will be able to identify and apply various knowledge representation methods for solving AI problems.

CO4: Students will be able to identify the applications of AI in the fields like Game playing and Expert systems.

CO5: Students will be able to differentiate AI and Machine learning, identify various Machine learning Techniques and apply ML in the context of supervised learning to solve real life problems.

Prerequisites:

- Fundamentals of Mathematics and Algorithms.

Course Outline:

Module 1: Introduction - History of Artificial Intelligence - problems, scope and applications, Characteristics of AI problems, problem space and search - production system- characteristics - structures and strategies for state space search, importance of state space to search and reasoning.

Module 2: Searching Techniques: Informed Searches and Uninformed searches, control and implementation of state space search, Heuristic searches - generate and test, hill climbing, Best–first search - A* algorithm, problem reduction, constraint satisfaction, means-ends analysis, heuristic in games, complexity issues.

Uninformed search: Depth First Search, Breadth First Search

Module 3: Knowledge Representation: Logic reasoning based on propositional logic and predicate logic representation and mappings, representing simple facts in logic, representing instances and ISA relationships, computable functions and predicates, resolution, natural deduction, knowledge representation using rules, forward versus backward reasoning, symbolic reasoning under uncertainty – non monotonic reasoning, logic programming.

Module 4: Game playing – the Minimax search procedure, adding Alpha-beta cutoffs, additional refinement, iterative deepening, planning system and its components.

Slot and filler structures: semantic nets, frames, conceptual dependency, scripts.

Expert Systems: Definition and characteristics of expert systems, representing and using domain knowledge, expert system shells. Knowledge engineering, knowledge acquisition, expert system life cycle & expert system tools, MCYCIN & DENDRAL examples of expert systems.

Module 5: Machine learning – Types of ML - Supervised learning, unsupervised learning and reinforcement learning, rote learning, Model development using Linear Regression, Model Visualization, Prediction and Decision Making, Model Evaluation: Over-fitting, Under-fitting and Model Selection.

Connectionist approaches: Artificial neural network, single neuron, activation functions, working of ANN. Deep learning and CNN.

References:

1. 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 Stubble field, 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.

MCS3C03:Object Oriented Programming Concepts

Course Number: 3.3
Hours per Week: 4
Credits: 4

Course Description:

The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Inheritance, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course helps the learners to develop Desktop GUI Applications, Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Course Outcomes:

CO1: Students will be able to explain the fundamental concepts of OOPS and will be able to create solutions for simple mathematical and scientific problems.

CO2: Students will be able to apply OOP concepts in writing java programs.

CO3: Students will be able to design multithreaded application programs in Java.

CO4: Students will be able to develop database application programs and GUI based programs.

CO5: Students will be able to develop web based applications by utilizing JSP concepts.

CO6: Students will be able to design software using OOP concepts and model using UML design diagrams.

Prerequisites:

- Knowledge in problem solving.

Course Outline:

Module 1: 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.

Module 2: 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

Module 3: 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, Exceptions - basic of Java exception handling, hierarchy, developing user defined exception classes.

Module 4: 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.

Module 5: 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.Phil Hanna , JSP 2.0: The Complete Reference
3. 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.
4. Grady Booch, James Rumbaugh and Ivar Jacobson, The Unified Modelling Language User Guide, 2nd Edition, Pearson, ISBN: 8131715825

MCS3L01: Practical III

Course Number: 3.6
Hours per Week: 4
Credits: 4

Part A - Course Description:

The purpose of this course is to study and exercise the sql queries for creating and maintaining databases. It includes creation, updation and deletion of databases and tables, modifying the structure of the tables, renaming, dropping of tables, adding different constraints and variety of data manipulations.

Course Outcomes:

CO1: Student will be able to design/create databases and tables using DDL commands.

CO2: Student will be able to insert data into tables using DML commands.

CO3: Student will be able to do different data manipulation and data retrieval operations using DML commands. Students will be able to implement functions and stored procedures using SQL.

CO5: Students will be able to design databases.

Course Outline:

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.
12. Consider the following schema for a Library Database:
    BOOK(Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS(Book_id, Author_Name) PUBLISHER(Name, Address, Phone) BOOK_COPIES(Book_id, Branch_id, No-of_Copies) BOOK_LENDING(Book_id, Branch_id, Card_No, Date_Out, Due_Date) LIBRARY_ BRANCH (Branch_id, Branch_Name, Address).
    Write SQL queries to
    (a) List the titles of all movies directed by 'Hitchcock'.
    (b) Find the movie names where one or more actors acted in two or more movies.
    (c) List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
    (d) Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
    (e) Update rating of all movies directed by 'Steven Spielberg' to 5
13. Consider the following schema for Order Database: SALESMAN(Salesman_id, Name, City, Commission) CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS(Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)
    Write SQL queries to:
    (a) Count the customers with grades above Bangalore's average.
    (b) Find the name and numbers of all salesman who had more than one customer.

(c) List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.)

(d) Create a view that finds the salesman who has the customer with the highest order of a day.

(e) Demonstrate the DELETE operation by removing salesman with id

14. Consider the schema for Movie Database:
ACTOR(Act_id, Act_Name, Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone) MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id) MOVIE_CAST(Act_id, Mov_id, Role) RATING(Mov_id, Rev_Stars)
Write SQL queries to:

(a) List the titles of all movies directed by 'Hitchcock'.

(b) Find the movie names where one or more actors acted in two or more movies.

(c) List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

(d) Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

(e) Update rating of all movies directed by 'Steven Spielberg' to 5.

15. Consider the schema for College Database:
STUDENT(USN, SName, Address, Phone, Gender) SEMSEC(SSID, Sem, Sec) CLASS(USN, SSID) SUBJECT(Subcode, Title, Sem, Credits) IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)
Write SQL queries to:

(a) List all the student details studying in fourth semester 'C'section.

(b) Compute the total number of male and female students in each semester and in each section.

(c) Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

(d) Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

(e) Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA< 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students

16. Consider the schema for Company Database:
EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo) DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate) DLOCATION (DNo,DLoc) PROJECT (PNo, PName, PLocation, DNo) WORKS_ON(SSN, PNo, Hours)
Write SQL queries to:

    (a) Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project.

    (b) Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise.

    (c) Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department.

    (d) Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).

    (e) For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000

PART B - Course Description:

This course aims to provide hands on experience on various object oriented concepts and their implementation through java programming language. It also enhances the capability of the students to design and implement solutions to real world computational problems.

- Experiments in Unit II are to be done using Java Programming language.
- Lab cycle for Unit II consists of 20 experiments.
- Minimum number of experiments in the Laboratory Record Book should be 15.
- Laboratory Record Book for Unit II should include Experiment No, Date, Aim of experiment, Class diagram, key method descriptions or algorithm, output and result.

Course Outcomes:

CO1: To apply the knowledge of java programming language for solving the simple real world computing problems

CO2: To create Java programs using inheritance and polymorphism.

CO3: To Implement error-handling techniques using exception handling and multi-threading

CO4: To create web programs using JSP technology

CO5: To implement GUI based applications using java swing

Course Outline:

1. Write a program to find the GCD of two numbers.
2. Write a program to print first n prime numbers.
3. Write a program to print first n Fibonacci numbers.
4. Write java program to check whether a given number is perfect, abundant or deficient.
5. Write a java program to count the number of even numbers, odd numbers, positive numbers, negative numbers and zeros in an array.
6. Write a java program to create a class Time with data member – hour, minute and second. Create two objects and find the sum of two times. Read the data members using constructor
7. Create a class Stack and write a program to implement stack operations in java.
8. Write a java program to find the area of shapes such as circle, rectangle and triangle using method overloading
9. Write a java program to illustrate method overriding
10. Write a java program to demonstrate inheritance hierarchy by using class 'shape' as the base class. Create two sub classes - 'triangle' and 'cube'. Write a program to find the area of the triangle and volume of cube.
11. Rewrite the above program to illustrate Dynamic method Dispatch.

12. Create an interface Calculator with methods to calculate sum, difference, product and quotient. Write a program to implement the above interface and display result.
13. Write a java program for generating two threads-Even and Odd. The odd thread display odd numbers and even thread display even numbers up to a limit.
14. Write a java program to illustrate user defined exceptions
15. Write a java program to write some text to a file and read it
16. Write simple client/server program using socket (Echo Server)
17. Write a program to draw line, rectangle and circle on an applet.
18. Write a swing application to accept two numbers and display their sum difference, product and quotient.
19. Write a JSP program to create an error page.
20. Write a JSP program to redirect a request to another JSP page.

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

Elective I

Elective I - MCS3E01 : List of Courses

Course		Title
No.	**Code**	
E1.1	MCS3E01a	Computer Graphics
E1.2	MCS3E01b	Introduction to Soft Computing
E1.3	MCS3E01c	Principles of Compilers
E1.4	MCS3E01d	Bioinformatics
E1.5	MCS3E01e	Computer Optimization Techniques
E1.6	MCS3E01f	Numerical and Statistical Methods

[5]Table Legend

[5]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS3E01a: Computer Graphics

Course Number: E1.1
Hours per Week: 4
Credits: 4

Course Description:

This course introduces fundamental concepts of Computer graphics and graphics programming. Various graphical devices that are used as display devices are discussed in the course. They will get a thorough understanding about various algorithms and operations applied in computer graphics. They will also be able to understand graphics programming based on the OpenGL graphics library.

Course Outcomes:

CO1: Students will be able to identify various applications and importance of Computer graphics and will be able demonstrate display working of various display devices.

CO2: Students will be able to demonstrate and apply various fundamental algorithms for 2D and 3D graphics primitive operations.

CO3: Students will be able to identify, demonstrate and apply 2D and 3D graphics transformations.

CO4: Students will be able to demonstrate and apply algorithms for object detection and backface removal.

CO5: Students will be able to demonstrate the understanding of OpenGL Graphics library and develop graphics programs using openGL.

C06: Students will be able to use OpenGL Graphics library for developing Graphics programs.

Prerequisites:

- Basic knowledge in Mathematics and C programming.

Course Outline:

Module 1: 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.

Module 2: 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

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

Module 4: 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.

Module 5: 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 Graphicswith Open GL, 4thEdition, 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, 6thEdition, Pearson Education, ISBN: 0132545233.

MCS3E01b: Introduction to Soft Computing

Course Number: E1.2
Hours per Week: 4
Credits: 4

Course Description:

This course provide ample opportunity to the students to learn basics of soft computing. This course provides a foundation on Fuzzy logic, Neural network and Evolutionary computing.

Course Outcomes:

CO1: students will be able to describe various approaches of soft computing

CO2: students will be able to apply genetic algorithms for problem solving

CO3: Students will be able to describe various neural network models for classification

CO4: Students will be able to compare and contrast fuzzy sets and ordinary sets and will be able to list out various applications of fuzzy sets.

CO5: Students will be able to describe various classification algorithms like SVM and evolutionary computing algorithms.

Prerequisites:

- Knowledge in fundamental mathematics.

Course Outline:

Module 1: 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.

Module 2: Introduction to genetic algorithm, genetic operators and parameters, genetic algorithms in problem solving, theoretical foundations of genetic algorithms, implementation issues – systems.

Module 3: 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

Module 4: Introduction to fuzzy sets, operations on fuzzy sets, fuzzy relations, fuzzy measures, applications of fuzzy set theory to different branches of science and engineering.

Module 5: 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.

MCS3E03c:Principles of Compilers

Course Number: E1.3
Hours per Week: 4
Credits: 4

Course Description:

This course provides an overview about the process of compilation and its design. The course covers the basic principles of all the modules in the compiler in general and covers lexical analysis, syntax analysis, intermediate code generation and code optimization in particular. An overview of region based analysis is also provided.

Course Outcomes:

CO1: Students will be able to identify and demonstrate the steps involved in the Compilation process.

CO2: Students will be able to identify the fundamental principles of lexical analysis.

CO3: Students will be able to compare various kinds of parsers.

CO4: Students will be able to classify various intermediate code representations and illustrate the translation of control flow statements.

CO5: Students will be able to explain the fundamental strategies in code generation and code optimization.

Prerequisites:

- Knowledge in data structures and programming.

Course Outline:

Module 1: 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.

Module 2: 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

Module 3: 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.

Module 4: 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.

Module 5: 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).

MCS3E01d: Bioinformatics

Course Number: E1.4
Hours per Week: 4
Credits: 4

Course Description:

This course provides an introduction to Bioinformatics and basic algorithms of computational biology. It also provides an introduction to tools used in Bioinformatics.

Course Outcomes:

CO1: Students will be able to explain nature and scope of computational biology and Bioinformatics

CO2: Students will identify various algorithms for Computational Biology

CO3: Students will be able to explain Sequence alignment.

CO4: Students will be able to explain Multiple sequence alignment.

CO5: Students will be able to use various tools for Bioinformatics.

Prerequisites:

- Basic knowledge in algorithm and biology.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: 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.

References:

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 andBioinformatics, Wiley-Interscience, ISBN: 9780471460541.
6. Richard M. Karp, Mathematical Challenges from Genomics and Molecular Biology, Notices of the American Mathematical Society, vol. 49, no. 5, pp. 544-553.
7. Glyn Moody, Digital Code of Life: How Bioinformatics is Revolutionizing Science, Medicine and Business, ISBN: 9780471327882.

8. 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.
9. Andrzej K. Konopka and M. James C. Crabbe, Compact Handbook of Computational Biology, 1st Edition, CRC Press, ISBN: 9780824709822.
10. Richard E. Bellman, Dynamic Programming, Princeton University Press, ISBN: 9780691146683.
11. 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.
12. Smith T F and Waterman M S, Identification of Common Molecular Subsequences, J. Mol. Bio. 147 (1981) 195–197.
13. Watson J D and Crick F H C, A Structure for Deoxyribose Nucleic Acid, Nature, 171 (1953) 737–738.
14. 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.

MCSE3E01e: Computer Optimization Techniques

Course Number: E1.5
Hours per Week: 4
Credits: 4

Course Description:

This course provide a foundation in Linear programming, transportation problem and dynamic programming.

Course Outcomes:

CO1: Students will be able demonstrate various Linear programming and sensitivity analysis

CO2: Students can apply Transportation and Network models for problem solving

CO3: Students will be able to apply simplex method and other Advanced linear programming methods fro problem solving

CO4: Students will be able to apply integer linear programming for problem solving

CO5: Students can apply Dynamic programming methods and Non linear programming methods for problem solving.

Prerequisites:

- Basic knowledge in Mathematics.

Course Outline:

Module 1: 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.

Module 2: Transportation and Network models - The transportation models and algorithm, the assignment and transhipment models, minimum spanning tree algorithm, shortest-route problem, maximum flow and min-cost models, critical path method and algorithms for matching.

Module 3: 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.

Module 4: Integer linear programming - illustrative applications, integer programming algorithms, uni-modularity and cutting-plane methods, travelling salesperson problem.

Module 5: 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: E1.6
Hours per Week: 4
Credits: 4

Course Description:

This is a foundation course in numerical methods.

Course Outcomes:

CO1: Students are able apply approximation and error computing methods

CO2: Students can solve linear equations.

CO3: Students are able to solve integration and Differential equations using approximation methods

CO4: Students can apply probability theories to solve problems

Prerequisites:

- Basic knowledge in mathematics and algorithms.

Course Outline:

Module 1: Approximation and errors in computing - introduction, significant digits - inherent errors – numerical error - modelling 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.

Module 2: 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.

Module 3: ntegration - 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.

Module 4: 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

Module 5: 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

Elective II - MCS3E02 : List of Courses

Course		Title
No.	**Code**	
E2..1	MCS3E02a	Pattern Recognition
E2.2	MCS3E02b	Wireless and Mobile Networks
E2.3	MCS3E02c	Cryptography and Network Security
E2.4	MCS3E02d	Advanced Web Technology
E2.5	MCS3E02e	Virtualization and Cloud Computing
E2.6	MCS3E02f	Data Warehousing and Data Mining

[6]Table Legend

[6]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS3E02a: Pattern Recognition

Course Number: E2.1
Hours per Week: 4
Credits: 4

Course Description:

This course provides a general introduction to pattern recognition. The steps involved in pattern recognition and various algorithms and approaches used in pattern recognition are discussed in this course.

Course Outcomes:

CO1: Students are able to identify basic steps in pattern recognition

CO2: Students are able to develop classification algorithms using statistical methods

CO3: Students are able to develop algorithms for supervised learning

CO4: Students are able to identify role of discriminant functions in pattern classification

CO5: Students are able apply unsupervised methods for pattern classification

Prerequisites:

- Basic knowledge in mathematics and algorithms.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: Linear discriminant functions - linear discriminant functions and decision surfaces – generalized linear discriminant functions - 2-category linearly separable case – nonseparable behaviour - linear programming algorithms, support vector machines - multilayer neural networks – feed forward operation and classification, back propagation algorithm, error surface, back propagation as feature mapping.

Module 4: Syntactic methods – stochastic search - Boltzmann learning – Nonmetric methods - decision trees – CART – other tree methods, grammatical methods, and grammatical inference.

Module 5: 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: E2.2
Hours per Week: 4
Credits: 4

Course Description:

This course provides a basic general overview of mobile networks. Various generations of wireless network, architecture of mobile network and security issues related to wireless mobile network are discussed here.

Course Outcomes:

CO1: Students are able identify the historic development of wireless communication

CO2: Students are able identify the open research problems in wireless communication

CO3: Students are able to identify various generations of cellular wireless networks

CO4: Students are able to identify various layers in mobile network

CO5: Students can describe various wireless network security issues.

Prerequisites:

- Basics of networking.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: Wireless network security – IEEE 80211i security – Wireless Transport Layer Security – sessions and connections – protocol architecture – WAP end-to-end security.

Module 5: 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.

MCS3E02c: Cryptography and Network Security

Course Number: E2.3
Hours per Week: 4
Credits: 4

Course Description:

This course introduces basics of computer security and cryptography. Various security issues and remedies are discussed in this course.

Course Outcomes:

CO1: Students are able identify security concepts, challenges and attacks in computer security

CO2: Students are able to principles and algorithms of Cryptography and cryptanalysis

CO3: Students are able to describe various message authentication methods

CO4: Students are able to apply various network security methods

CO5: Students are able to identify various network intruders and are able to set up firewalls for security purposes

Prerequisites:

- Knowledge in computer networks and operating systems.

Course Outline:

Module 1: 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.

Module 2: Message authentication – approaches – MAC – one way Hash function – secure Hash functions – Message Authentication Codes. Public key cryptography principles – algorithms – digital Signatures.

Module 3: 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.

Module 4: 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.

Module 5: 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: E2.4
Hours per Week: 4
Credits: 4

Course Description:

This course introduces advanced web programming technology and techniques. This course also provide Python introduction and how it is used for srver side scripting. Also, this course gives a idea about using databases with python.

Course Outcomes:

CO1: Students are able to identify characteristics and technologies

CO2: Students are able to describe various web services architecture.

CO3: Students are able to install and use python for developing programs

CO4: Students are able to develop server side programs using python

CO5: Students are able to connect SQLite with python and perform SQL operations

Prerequisites:

- Basics of programming.

Course Outline:

Module 1: 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.

Module 2: 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).

Module 3: 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

Module 4: 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

Module 5: 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: E2.5
Hours per Week: 4
Credits: 4

Course Description:

This course introduces cloud computing and virtualization. Infrastructure, programming model and security issues related to cloud are discussed in this course

Course Outcomes:

CO1: Students are able to describe various cloud computing models , architecture and infrastructures

CO2: Students are able implement various levels of virtualization in cloud computing

CO3: Students are able to use Cloud software environments and infrastructures.

CO4: Students are able to demonstrate various distributed programming methods and models.

CO5: Students are able to describe various security issues and challenges in the cloud computing environments

Prerequisites:

- Basic knowledge in Networking and programming

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: 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. Tom White, Hadoop : The Definitive Guide, O'Reilly Media, ISBN: 9780596551360.

MCS3E02f: Data Warehousing and Data Mining

Course Number: E2.6
Hours per Week: 4
Credits: 4

Course Description:

The course deals with data warehousing and data mining and covers important topics including, data warehouse architecture, OLAP operations, data mining fundamentals, association rule mining, classification, clustering and other latest mining techniques.

Course Outcomes:

CO1: Students will be able to summarize the fundamental concepts of data warehousing like data warehouse architecture, OLAP operations, data warehouse schemas and multi-dimensional data representation

CO2: Students will be able to outline the activities in data pre-processing

CO3: Students will be able to illustrate and compare various association rule mining algorithms.

CO4: Students will be able to compare various classification algorithms and apply them in designing data mining applications

CO5: Students will be able to summarize different clustering approaches, outlier mining algorithms and explain the concepts of other modern mining techniques

Prerequisites:

- Basic knowledge in programming and mathematics

Course Outline:

Module 1: 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.

Module 2: Data mining – introduction – definition - data mining functionalities – major issues in data mining - data pre-processing – data cleaning – data integration and transformation – data reduction – data discretization and concept hierarchy generation.

Module 3: Association rule mining - efficient and scalable frequent itemset mining methods – mining various kinds of association rules – association mining to correlation analysis – constraint-based association mining.

Module 4: 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 – lazy learners – other classification methods – prediction – accuracy and error measures – evaluating the accuracy of a classifier or predictor – ensemble methods – model section

Module 5: 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.

Introduction to other mining techniques (concepts only)-Graph mining, spatial mining, multimedia data mining, text mining and web data mining.

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.

Fourth Semester

Semester IV

Course		Title	C	Mark Weightage			Hours Per Week		
No.	**Code**			**I**	**E**	**T**	**L**	**P**	**T**
4.1	MCS4E03	Elective III	3	1	4	5	5	0	5
4.2	MCS4E04	Elective IV	3	1	4	5	5	0	5
4.3	MCS4P01	Project Requirement Analysis and Design – Related Discussion	8	1	4	5	3	1	4
		Project Coding, Testing & Implementation - Related Discussion					2	2	4
		Project Evaluation and Assessment					2	0	2
		Project Lab Work					0	5	5
		Total	**14**				**17**	**8**	**25**

[7]Table Legend

[7]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS4P01: Project Work

Course Number: 4.1
Hours per Week: 15
Credits: 8

Course Description:

This course indents to provide the students ample opportunities for putting all the knowledge gathered from the entire programme into practice and come up with a workable solution for an identified problem in any domain or to come up with research findings.

Course Outcomes:

CO1: Students are able to apply their knowledge in various courses for developing software or producing new knowledge in the form of a research paper/findings.

CO2: Students are able to verify an existing knowledge.

Prerequisites:

- Knowledge gathered from the entire courses.

Course Outline:

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

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 real-life application oriented project work or research and development projects. The student can formulate a project problem with the help of her/his guide and submit the project proposal of the same. Approval of the project proposal is mandatory. If approved, the student can commence working on it, and complete it.

Guidelines for Submission of Project Report:

The distinguishing mark of a dissertation is an original contribution to the knowledge. The dissertation is a formal document whose sole purpose is to prove that the student have made an original contribution to the knowledge. Failure to prove that the student have made such a contribution generally leads to failure in the programme.

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, analysis and arguments but 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. Above all it should be *free from the plagiarism*.

The dissertation is to be prepared in LaTeXformat (either or a suitable Windows LaTeX variant). 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.

Elective III

Elective III- MCS4E03 : List of Courses

Course		Title
No.	**Code**	
E3.1	MCS4E03a	Data Compression
E3.2	MCS4E03b	Pervasive Computing
E3.3	MCS4E03c	System Security
E3.4	MCS4E03d	Molecular Simulation and Modelling
E3.5	MCS4E03e	Fundamentals of Big Data
E3.6	MCS4E03f	Web Engineering

[8]Table Legend

[8]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS4E03a: Data Compression

Course Number: E3.1
Hours per Week: 4
Credits: 4

Course Description:

This course introduces the basic applications, concepts, and techniques of Data Compression and develops skills for using recent data compression software to solve practical problems in a variety of disciplines.

Course Outcomes:

CO1: Students are able to demonstrate basic data compression algorithms

CO2: Students are able to describe the physical significance of some basic concepts of information theory including entropy, average mutual information and the rate distortion bound.

CO3: students are able to design entropy codes including Huffman codes and arithmetic coding.

CO4: Students are able to demonstrate the operation of lossless compression schemes.

CO5: Students are able to apply various transformations techniques for compression

CO6: Student will be able to develop a reasonably sophisticated data compression application.

Prerequisites:

- Knowledge in mathematics.

Course Outline:

Module 1: Introduction Mathematical Preliminaries, Lossy and Lossless compression, Application of compression. Simple lossless encoding - Run length encoding, Huffman coding, LZW coding, Run length encoding, Arithmetic coding.

Module 2: Fundamentals of Information Theory-Concepts of entropy, probability models, markov models, Fundamentals of coding theory, Algorithmic information, theory Minimum description.

Module 3: 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, Lossless Compression standards, zip, gzip, bzip, unix compress, GIF, JBIG.

Module 4: 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.

Module 5: 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: E3.2
Hours per Week: 4
Credits: 4

Course Description:

This course introduces applications, devices, technologies and architecture for pervasive computing.

Course Outcomes:

CO1: Students are able to identify need, application and challenges of pervasive computing

CO2: Students will be able to propose appropriate technology to achieve the pervasive computing

CO3: Students are able to explain web architectures for pervasive computing.

CO4: Students are able to describe web voice technologies

Prerequisites:

- Basics of networking and internet.

Course Outline:

Module 1: 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.

Module 2: Device technology for pervasive computing - hardware, human-machine interfaces, biometrics, operating systems, Java for pervasive devices, outlook. Device connectivity - protocols, security, device management.

Module 3: 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.

Module 4: 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.

Module 5: 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: E3.3
Hours per Week: 4
Credits: 4

Course Description:

This course introduces various security aspects of computer. Different levels of security and how to achieve and manage security are focused in this course.

Course Outcomes:

CO1: Students are able to demonstrate their understanding of the differences between various forms of computer security, where they arise, and appropriate tools to achieve them.

CO2: Students will be able to propose appropriate tools to achieve the desired level of security.

CO3: Students are able to demonstrate their understanding on operating system security

CO4: Students are able to describe database security

CO5: Students are able to administrate the security

Prerequisites:

- Basics of operating system, networking and internet.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: Database Security - security requirements - integrity of database, confidentiality and availability, reliability and integrity, sensitive data, interface, multilevel database, proposals for multilevel security.

Module 5: 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.

MCS4E3d: Molecular Simulation and Modelling

Course Number: E3.4
Hours per Week: 4
Credits: 4

Course Description:

This course gives an overview of molecular modelling. Various approaches and methods for modelling and gene mapping are given in the course.

Course Outcomes:

CO1: Students will be able to apply simulation techniques to study molecular dynamics.

CO2: Students will be able to learn and apply statistical approaches and models for phylogenetic analysis. understand and apply docking simulations.

CO3: Students are able perform Computational gene mapping

CO4: Students can do molecular visualization

CO5: Students are able to describe structural models of protiens.

Prerequisites:

- Basics of Biology and programming.

Course Outline:

Module 1: Overview of molecular modelling - molecular modelling methods - semi-empirical method and empirical method. Model Type - static, dynamic and probabilistic models. Models of growth and decay.

Module 2: System modelling - concept, principles of mathematical modelling, 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.

Module 3: 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.

Module 4: Structural Modelling: 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 modelling. 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.

Module 5: 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, and advantages of Swiss-PdbViewer.

References:

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

MCS4E3e: Fundamentals of Big Data

Course Number: E3.5
Hours per Week: 4
Credits: 4

Course Description:

The course deals with data warehousing and data mining and covers important topics including, data warehouse architecture, OLAP operations, data mining fundamentals, association rule mining, classification, clustering and other latest mining techniques.

Course Outcomes:

CO1: Students will be able to summarize the fundamental concepts of big data, big data architecture, role of CMS and big data management

CO2: Students will be able to outline the processes in big data analytics and various big data analysis frameworks

CO3: Students will be able to summarize the fundamentals of NOSqL and the architecture and operations in MongoDB

CO4: Students will be able to explain HDFS and application development in Hadoop

CO5: Students will be able to write simple MapReduce programs, and handle UFO data

Prerequisites:

- Knowledge of Database Management Systems.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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 - implementing index-related functions - min() and max().

Module 4: 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.

Module 5: 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 E nterprise 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: E3.6
Hours per Week: 4
Credits: 4

Course Description:

This course deals with the various aspects like architecture, modelling, applications and testing of web engineering.

Course Outcomes:

CO1: Students are able describe, web engineering, categories and characteristics of web applications

CO2: Students are able describe Requirements Engineering

CO3: Students are able demonstrate web application architectures

CO4: Students are able to model web applications

CO5: Students are able to test web applications

Prerequisites:

- Knowledge in software engineering and web development.

Course Outline:

Module 1: Web Engineering (WE) – introduction – motivation – categories characteristics of web applications – product related, usage related and development related – evolution of WE.

Module 2: 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.

Module 3: 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.

Module 4: Modelling web applications – introduction – modelling specifics in WE – levels – aspects – phases of customizations – modelling requirements – hypertext modelling - hypertext structure modelling concepts – access modelling 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 – designing a link representation – designing link internals – navigation and orientation – structural dialog for complex activities – interplay with technology and architecture – functional design.

Module 5: 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

Elective IV- MCS4E04 : List of Courses

Course		Title
No.	**Code**	
E4.1	MCS4E04a	Digital Image Processing
E4.2	MCS4E04b	Introduction to Block chain technology
E4.3	MCS4E04c	Software Development for Portable Devices
E4.4	MCS4E04d	Storage Area Networks
E4.5	MCS4E04e	Semantic Web
E4.6	MCS4E04f	Advanced Java Programming

[9]Table Legend

[9]C: Credits, I: Internal Component (%), E: External Component (%), L: Lecture Hours, P: Practical Hours, T: Total Hours

MCS4E04a: Digital Image Processing

Course Number: E4.1
Hours per Week: 4
Credits: 4

Course Description:

This course is intended to give fundamentals of digital image processing. The course will cover various transformations, enhancement techniques and restoration techniques used in image processing.

Course Outcomes:

CO1: Students are able to demonstrate the basics of image processing and its understanding.

CO2: Students are able to demonstrate various transformation techniques used in image processing

CO3: Students are able to demonstrate and apply various image enhancement techniques using filters.

CO4: Students are able to demonstrate various noise models in image processing

CO5: Students are able demonstrate and apply image compression techniques

Prerequisites:

- Basic knowledge of mathematics.

Course Outline:

Module 1: 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.

Module 2: Image transforms - introduction to Fourier transform - discrete Fourier transform (DFT) - properties DFT- other separable image transforms - Walsh, Hadamard and Discrete Cosine transforms. Hotelling transform.

Module 3: 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.

Module 4: 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.

Module 5: 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: Advanced Database Design

Course Number: E4.2
Hours per Week: 4
Credits: 4

Course Description:

This course provides advanced database design concepts. EER, OODBMS are also dealt here.

Course Outcomes:

CO1: Students are able design database using ER EER models

CO2: Students are able demonstrate their understanding on OODBMS

CO3: Students are able to design databases using OO concepts

CO4: Students are able to describe Parallel and distributed databases and client-server architecture

CO5: Students are able to use Object databases on the web and semi structured data

Prerequisites:

- Basics of DBMS and OOPs.

Course Outline:

Module 1: The Extended Entity Relationship model and object model - The ER model revisited, motivation for complex data types, user defined abstract data types and structured types, subclasses, super classes, inheritance, specialization and generalization, constraints and characteristics of specialization and generalization, relationship types of degree higher than two.

Module 2: Object-Oriented databases - overview of object-oriented concepts, object identity, object structure, and type constructors, encapsulation of operations, methods, and persistence, type hierarchies and inheritance, type extents and queries, complex objects, database schema design for OODBMS, OQL, persistent programming languages, OODBMS architecture and storage issues, transactions and concurrency control, example of ODBMS.

Module 3: Object relational and extended relational databases - database design for an ORDBMS - nested relations and collections, storage and access methods, query processing and optimization, an overview of SQL3, implementation issues for extended type - systems comparison of RDBMS, OODBMS and ORDBMS.

Module 4: Parallel and distributed databases and client-server architecture - architectures for parallel databases, parallel query evaluation, parallelizing individual operations, sorting, joins, distributed database concepts, data fragmentation, replication and allocation techniques for distributed database design, query processing in distributed databases, concurrency control and recovery in distributed databases. An overview of client-server architecture.

Module 5: Object databases on the web and semi structured data - web interfaces to the web, overview of XML - structure of XML data, document schema, querying XML data - storage of XML data, XML applications - the semi structured data model, implementation issues, indexes for text data. Enhanced data models for advanced applications - active database concepts, temporal database concepts, spatial databases concepts and architecture, deductive databases and query processing, mobile databases, geographic information systems.

References:

1. Elmasri and Navathe, Database Systems – Models, Languages, Design and Application Programming, 6th Edition, Pearson India, ISBN: 8131792471.
2. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw - Hill Education, ISBN: 9339213114.
3. Korth, Silberchatz and Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill Education India Pvt. Ltd, ISBN: 9332901384.
4. Alexis Leon and Mathews Leon, Database Management System, 1st Edition, Vikas Publishers, ISBN: 8182092221.
5. Peter Rob and Coronel, Database Systems, Design, Implementation and Management, 5th Revised Edition, Course Technology, ISBN: 061906269X.

6. C J Date, Introduction to Database Systems, 8th Edition, Addison-Wesley, ISBN: 0321197844

MCS4E04c: Software Development for Portable Devices

Course Number: E4.3
Hours per Week: 4
Credits: 4

Course Description:

This course introduces various tools and techniques used for the development of mobile based application programs.

Course Outcomes:

CO1: Students are able to apply HTML 5 and CSS3 for Mobile web

CO2: Students are able apply jQuery for web development

CO3: Students are able to demonstrate Android architecture

CO4: Students are able to use databases for mobile app development

CO5: Students are able to use various connectivity techniques for mobile web.

Prerequisites:

- Basics of networking and internet.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: 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

MCS4E04d: Storage Area Networks

Course Number: E4.4
Hours per Week: 4
Credits: 4

Course Description:

This course will provide an insight into Storage Area Networks, its architecture and management. It also helps get an idea about designing and building SAN.

Course Outcomes:

CO1: Students will be able to outline the basic networking concepts and topologies

CO2: Students will be able to summarize the SAN fundamentals

CO3: Students will be able to describe the Storage networking architecture and list the emerging SAN interconnect technologies

CO4: Students will be able to summarize Storage network management

CO5: Students will be able to describe how to design and build a SAN

Prerequisites:

- Knowledge in networking and hardware concepts.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: 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: 1
Hours per Week: 4
Credits: 4

Course Description:

This course introduces various tool, techniques and algorithms for semantic web. Applications, security issues and current trends in the semantic web are also given focus.

Course Outcomes:

CO1: Students are able describe architecture for semantic web

CO2: Students are able to use languages for semantic web

CO3: Students are able to demonstrate various ontology learning algorithms used in semantic web

CO4: Students are able use management tools for semantic web

CO5: Students are able identify various application, security issues, and current trends in semantic web

Prerequisites:

- Basics of web applications.

Course Outline:

Module 1: Components – types – ontological commitments – ontological categories – philosophical background – knowledge representation ontologies – top level ontologies – linguistic ontologies – domain ontologies – semantic web – need – foundation – layers – architecture.

Module 2: 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.

Module 3: 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.

Module 4: 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

Module 5: 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: E4.6
Hours per Week: 4
Credits: 4

Course Description:

This course indented for giving a thorough understanding about advanced Java programming concepts. It provides details of RMI, EJB and JSP

Course Outcomes:

CO1: Students are able to describe RMI and Servlet architectures

CO2: Students are able configure Tomcat Server and use servlets to manipulate response headers.

CO3: Students are able to demonstrate JNDI & EJB architectures

CO4: Students are able to use JSP for developing web pages

CO5: Students are able to apply JSP for server side programming.

Prerequisites:

- Basics of Java programming.

Course Outline:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: Hibernate - 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.

Appendix A

Detailed Program List

A.1 MCS1L01 - Practical I - PART A

Detailed Program List - C Programming

1. To swap the values of two variables
2. To compute the area of a triangle
3. To find the biggest of two numbers using simple if statement and if else statement
4. To run a simple calculator that accepts two operands and an operator such as addition and subtraction. Implement using simple if statement, if else statement, else if ladder and switch.
5. To display digits of an integer in the reverse order. Implement using for, while and do while statements.
6. To check whether a given number is prime or not. Implement using for, while and do while statements.
7. Design a program to find the frequency of different vowels and special characters such as , â̂nd white spaces in a paragraph. Use switch statement.
8. To find the factorial of a given number. Implement using for, while and do while statements.
9. To count the number of positive and negative numbers entered by the user. Program should print the count of positive and negative numbers and terminate, when the user inputs 999. When 0 is accepted, control should go to the

beginning of the loop (Implement termination and branching back to the beginning of loop using goto statement and repeat the same using break and continue statements).

10. To accept n numbers into an integer array and copy its contents to another array.
11. To accept n numbers into an integer array and print the elements in the reverse order.
12. To accept n numbers into an integer array and search for an element in the array (use linear search).
13. To accept n numbers into an integer array and erase duplicate elements in the array.
14. To input a square matrix and find its transpose and trace.
15. To accept two 2D matrices and display their sum. Implement using array and pointer notations.
16. To accept two 2D matrices and display their product. Implement using array and pointer notations.
17. To print the following pattern:

```
1

232

34543

4567654

567898765
```

18. To count the occurrence of a particular word in the string.
19. To insert a substring into a string.
20. To compute the sum of the array elements using pointers.
21. To sort a list of N names in ascending order. Implement using 2D arrays and a character array of pointers to store names.

22. To find the length of a string. Write a user defined function to do this task.
23. To compare two strings. Write a user defined function to do this task.
24. To find the factorial of a number (use recursion).
25. To find the fibonacci series upto a given term (use recursion).

A.2 MCS2L01 - Practical II - PART B

Detailed Program List - Data Analysis using Python

1. Find sum of n numbers.
2. Find a average of n numbers
3. Check a number is palindrome or not.
4. Find largest and minimum number in an array using python
5. Develop a simple & funny prediction game.
6. Generate prime numbers up-to a limit.
7. Find factorial using recursion.
8. Add two matrices.
9. Generating random numbers, Random Float, Random Array.
10. Generate a 2-D array with 3 rows, each row containing 5 random integers from 0 to 100
11. Multiply two Matrices using Numpy.
12. Finding maximum and minimum of array using NumPy
13. Finding Mean,Median, Standard Deviation and Variance using NumPy
14. Sort an array using Numpy.
15. Use pyplot to draw Line Plot, Bar Plot, Scatter Plot
16. Plotting 3-D Lines and Points.
17. Plot a pie chart in Python using Matplotlib.

18. Draw Sine Wave Plot using Numpy.
19. Use describe() method to provide a quick overview of the numerical data in a DataFrame.
20. Read Titanic data set using pandas.
21. Make all name characters lowercase in Titanic data set
22. Create a new column Surname that contains the surname of the passengers by extracting the part before the comma.
23. Create a series from array, list and dictionary
24. Reading data from CSV and JSON files
25. Plotting a Line Chart using Pandas.

Appendix B

Project Report Template

<<PROJECT/THESIS REPORT TEMPLATE>>

A Thesis submitted

By

<<Student Name>>

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
Farook College P.O, Kozhikode
Affiliated to University of Calicut
Kerala, 673 632
INDIA.

<<MONTH YEAR>>

FAROOK COLLEGE (AUTONOMOUS)
FAROOK COLLEGE P.O, KOZHIKODE
AFFILIATED TO UNIVERSITY OF CALICUT
KERALA, 673 632
INDIA.

CERTIFICATE

This is to certify that the thesis entitled "<<**Project/Thesis Report Template**>>" is a report of original work carried out by <<**Student Name**>> under my supervision and guidance in the Department of Computer Science, Farook College(Autonomous), Kozhikode , and that no part there of has been presented for the award of any other degree.

FAROOK COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE

DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the supervision of <<Prof./ Dr./Mr Guide's Name>> in the Department of Computer ScienceFarook College (Autonomous), and that no part thereof has been presented for the award of any other degree.

<<**Student Name**>>
Department of Computer Science
Farook College(Autonomous), Kozhikode

Table of Contents

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.

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.

Acknowledgements

Your Acknowledgement Goes here

Farook College
Month Day, Year

<<Studnet Name >>

Chapter 1

INTRODUCTION

1.1 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 – Title, 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

SYSTEM DESIGN

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,
  - lack of documentation;
  - lack of suitable supporting software;
  - 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.

- Brief description about the Tools/Scripts for Implementation
- Module Hierarchy
- Coding
- Problems Encountered

Chapter 6

TESTING

This chapter includes the following subsections.

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

Bibliography