Proposal for

Revised Syllabus

For

Master of Science (MSc) – Computer Science
M.Sc. Computer Science Programme Structure (Proposal)  
(2014 - Admission Onwards)

This document contains the draft of the proposed syllabus for MSc Computer Science for the academic year 2014-15 in University Centres and affiliated colleges. The complete draft syllabus is enclosed herewith for your information and review.

Please thoroughly go through the scheme\contents and inform me your feedbacks at the earliest. This will help me to take necessary steps for incorporating the required changes before placing it in the Board of Studies meeting.

The copy of your feedbacks may also be copied to: abdulhaleem@farookcollege.ac.in

Expecting a prompt response in this regard.

NB: Please circulate this information to all concerned faculty members, students and subject experts in your contact list.

With warm regards,

Dr. LAJISH.V.L  
Assistant Professor & Head (i/c) &  
Chairman, PG Board of Studies in CS & Applications.  
Department of Computer Science  
University of Calicut, Kerala-673635, INDIA  
E-mail: lajish@uoc.ac.in  
Tel: +91-494-24017325 (Office)  
: +91-9495793094 (Cell)  
Web:  http://www.universityofcalicut.info
1. Course should be helpful for the students
   a. To enable them to contest for competitive examinations like NET/GATE/JEST etc.
   b. to take up a research oriented higher course such as MPhil/PhD, Or
   c. to help them to secure a position in the industry.

2. These objectives are desired by giving the students choices for their “specialization” through a set of electives right from the Semester II.

3. You are kindly request to contribute your ideas in the following aspects:
   a. Are there any papers to be included/excluded in/from the core papers section?
   b. Can even the core papers be made industry relevant by adding topics relevant to the present. If yes, portions to be added to each of such subjects.
   c. Are the broad range of topics proposed under Electives (I, II, III and IV) are sufficient to enable the student to decide his specialization? – these subjects can be conventional/industry related/ or a mix of both.

4. Kindly go through this document and check the following:
   a. Whether the course has a logical flow right from the first semester to the last semester?
   b. Does the syllabus proposed for each course reflect the intended objectives (overall and course wise)?
   c. Are the syllabus contents realistic – can they be covered within one semester?
   d. Are they any overlapping of contents in the syllabus (for eg, between Theory of Computation and Discrete Structures)?
   e. Does the syllabus reflect the contents of the reference/text books listed?
   f. Are the text books readily available?
   g. Practical
      i. Is the list of experiments and corresponding theory portions have a correspondence each other?
      ii. Is the list sufficient?
      iii. Are there any experiments that are not “viable” or not suitable?

5. In case, if you have a modification to be proposed for any of the subjects, please use the attached format to specify your proposals (Proposal For Modification of Existing Subjects.doc).

6. If you would like to propose a new course, that can also be done in the attached format (Proposal For Subject.doc).
### Course Structure, Scheme of Evaluation

#### Semester I

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Instructional Hours/week</th>
<th>Marks</th>
<th></th>
<th></th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1.1</td>
<td>Discrete Mathematical Structures</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>S1.2</td>
<td>Advanced Data Structures</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>S1.3</td>
<td>Theory of Computation</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>S1.4</td>
<td>The Art of Programming Methodology</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>S1.5</td>
<td>Computer Organization &amp; Architecture</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>S1.6</td>
<td>Practical 1 (1.2 &amp; 1.4) - (Departmental)</td>
<td>0</td>
<td>4</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>24</strong></td>
</tr>
</tbody>
</table>

#### Semester II

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Instructional Hours/week</th>
<th>Marks</th>
<th></th>
<th></th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>S2.1</td>
<td>Design and Analysis of Algorithms</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>S2.2</td>
<td>Operating System Concepts</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>S2.3</td>
<td>Computer Networks</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>S2.4</td>
<td>Computational Intelligence</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>S2.5</td>
<td>Elective I</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>S2.6</td>
<td>Practical 2 (S2.2 &amp; S2.3) (University)</td>
<td>0</td>
<td>4</td>
<td>25</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>S2.7</td>
<td>Seminar</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>25</strong></td>
</tr>
</tbody>
</table>
Elective II S2.5 - List of Courses

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>11a</td>
<td>S2.5a</td>
<td>Computer Graphics</td>
</tr>
<tr>
<td>11b</td>
<td>S2.5b</td>
<td>Introduction to Soft Computing</td>
</tr>
<tr>
<td>11c</td>
<td>S2.5c</td>
<td>Web Technology</td>
</tr>
<tr>
<td>11d</td>
<td>S2.5d</td>
<td>Bio Informatics</td>
</tr>
<tr>
<td>11e</td>
<td>S2.5e</td>
<td>Computer Optimization Techniques</td>
</tr>
<tr>
<td>11f</td>
<td>S2.5f</td>
<td>Numerical and Statistical Methods</td>
</tr>
</tbody>
</table>

Semester III

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Instructional Hours/week</th>
<th>Marks</th>
<th>Total Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>S3.1</td>
<td>Advanced Data Base Management System</td>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>S3.2</td>
<td>Principles of Compilers</td>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>S3.3</td>
<td>Object Oriented Programming Concepts</td>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>S3.4</td>
<td>Elective II</td>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>S3.5</td>
<td>Elective III</td>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>S3.6</td>
<td>Practical 3 (S3.1 &amp; S3.3) (Departmental)</td>
<td>0</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

Elective II S3.4 – List of Courses

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>17a</td>
<td>S3.4a</td>
<td>Pattern Recognition</td>
</tr>
<tr>
<td>17b</td>
<td>S3.4b</td>
<td>Wireless and Mobile Networks</td>
</tr>
<tr>
<td>17c</td>
<td>S3.4c</td>
<td>Cryptography &amp; Network Security</td>
</tr>
<tr>
<td>17d</td>
<td>S3.4d</td>
<td>Advanced Web Technology</td>
</tr>
<tr>
<td>17e</td>
<td>S3.4e</td>
<td>Virtualisation And Cloud Computing</td>
</tr>
<tr>
<td>17f</td>
<td>S3.4f</td>
<td>Data Warehousing and Data Mining</td>
</tr>
</tbody>
</table>
### Elective III S3.5 – List of Courses

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>18a</td>
<td>S3.5a</td>
<td>Data Compression</td>
</tr>
<tr>
<td>18b</td>
<td>S3.5b</td>
<td>Pervasive Computing</td>
</tr>
<tr>
<td>18c</td>
<td>S3.5c</td>
<td>System Security</td>
</tr>
<tr>
<td>18d</td>
<td>S3.5d</td>
<td>Molecular Simulation and Modeling</td>
</tr>
<tr>
<td>18e</td>
<td>S3.5e</td>
<td>Fundamentals of Big Data</td>
</tr>
<tr>
<td>18f</td>
<td>S3.5f</td>
<td>Web Engineering</td>
</tr>
</tbody>
</table>

### Semester IV

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Instructional Hours/week</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>S4.1</td>
<td>Elective IV</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>S4.5</td>
<td>Major Project + SE &amp; RM (Duration of the Project = 16 Weeks) (University)</td>
<td>Duration of the Project = 16 Weeks Total Theory Hours for SE &amp; RM = 20 Hours</td>
<td>300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Internal</th>
<th>Final</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>75</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>100</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

**Total Credits 12**

### Elective IV S4.1 – List of Courses

<table>
<thead>
<tr>
<th>Course No</th>
<th>Subject Code</th>
<th>Subject Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>19a</td>
<td>S4.1a</td>
<td>Digital Image Processing</td>
</tr>
<tr>
<td>19b</td>
<td>S4.1b</td>
<td>Advanced Topics in Database Design</td>
</tr>
<tr>
<td>19c</td>
<td>S4.1c</td>
<td>Software Development for Portable Devices</td>
</tr>
<tr>
<td>19d</td>
<td>S4.1d</td>
<td>Storage Area Networks</td>
</tr>
<tr>
<td>19e</td>
<td>S4.1e</td>
<td>Semantic Web</td>
</tr>
<tr>
<td>19f</td>
<td>S4.2f</td>
<td>Advanced Java Programming</td>
</tr>
</tbody>
</table>
M.Sc. Computer Science

SYLLABUS

First Semester

S1.1 Discrete Mathematical Structures

Course Number: 1

Prerequisites/ Exposure: None

Objectives: To introduce discrete mathematics concepts necessary to understand basic foundation of computer science.


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


Unit V: Graph Theory - Paths, Cycles and Connectivity, Subgraphs, Types of Graphs, Representation of Graphs, Graph Isomorphism, Bipartite Graphs, Subgraphs, Eulerian and Hamiltonian Graphs. Trees - Spanning Trees, Cayley's theorem. Prim's and Kruskals Algorithm - Shortest Path Problem - Dijkstra's Algorithm.

Page | 7
References:
1. J.K. Sharma, Discrete Mathematics, Macmillan India Ltd.

S1.2 Advanced Data Structures
Course No: 2

Prerequisites/ Exposure: None

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

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

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


References:


S1.3 Theory of Computation
Course No: 3

Prerequisites/Exposure: None

Objectives: To provide students with an understanding of basic concepts in the theory of computation.

Unit I: Preliminaries - Introduction to formal proof and inductive proofs - The central concepts of Automata Theory - Alphabets, Strings, Languages - Introduction to automata and grammar - Deterministic Finite Automata, Non-deterministic Finite Automata - Equivalence of Deterministic and Nondeterministic Finite Automata - Finite Automata with Epsilon Transitions - Equivalence of NFA with and without epsilon moves.
**Unit II**: Regular Expressions, Finite Automata and Regular Expressions, Properties of regular Languages - Pumping lemma and proof for existence of non regular languages, Closure properties, homomorphism, substitution - Decision Properties - Equivalence and Myhill Nerode and DFA state minimization - Regular Grammar.

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

**Unit IV**: Turing machines - TM computations - Equivalence of standard TM with multi tape and non deterministic TM’s - Turing acceptable, Turing decidable and Turing enumerable language classes - Equivalence of type 0 grammars with TM’s - Church’ thesis - Chomsky hierarchy - Closure properties of recursive and recursively enumerable languages.

**Unit V**: Computability and Decidability - halting problem - reductions - post correspondence problem. Computational complexity - Time and space bounded simulations - Classes P and NP – NP completeness - Cook’s theorem.

**References**

3. Linz P, An Introduction to formal Languages and Automata, Narosa.
**S1.4 The Art of Programming Methodology**

Course No: 4

Prerequisites/Exposure: None

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


**Unit II:** Introduction to C Programming, overview and importance of C, C Program Structure and Simple programs, Creation and Compilation of C Programs under Linux and Windows Platforms. Elements of C Language and Program constructs: - structure of C program - character set, tokens, keywords, identifier - Data types, constants, symbolic constants, variables, declaration, data input and output, assignment statements. Operators in C - arithmetic operators, relational operators, logical operators, assignment operators, increment and decrement operators, conditional operators, special operators, precedence of operators - arithmetic expressions - evaluation of expressions, type conversion in expressions - precedence and associativity - mathematical functions - I/O operations.

**Unit III:** Decision making - IF statement, IF ELSE statement, Nesting of IF ELSE and ELSE IF Ladder, SWITCH statement, BREAK statement, CONTINUE statement, GOTO statement, return statement - Looping - WHILE, DO-WHILE, and FOR loops, nesting of loops, skipping & breaking loops - Arrays - single dimension arrays - accessing array elements - initializing an array, two dimensional & multi dimensional arrays - memory representation - strings - processing of strings - string manipulation functions.

**Unit IV:** The Concept of modularization - 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 and recursion - passing arrays & strings to function - returning multiple values - recursion - scope and life time of variables storage class specifiers - automatic, extern, static storage, register storage - Structures & Union definition, giving values to members, structure initialization, comparison of structure variables, arrays of structures, arrays within structures, structures within arrays, structures and functions, Unions, bit-fields.

**Unit V:** Pointer - pointer operator - pointer expression - declaration of pointer - initializing pointer - de-referencing - pointer to pointer, constant pointer, array of pointers, pointer to function. Files - file handling - defining & opening a file - closing a file - Input/output operations on files - error handling, random access to files, command line arguments - dynamic memory allocation - preprocessor directives: macro substitution directives - simple macros - macros with arguments - nesting of macros, compiler control directives.

**References**

**S 1.5 Computer Organization & Architecture**

Course No: 5

**Prerequisites/ Exposure:** None

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

**Unit I:** Number systems and Conversions, Boolean Algebra - Truth Tables - Logic gates and Map simplification - flip-flops - design of combinational and sequential circuits - examples of digital circuits - adders, multiplexers, decoders, counters, shift registers - register transfer language and micro operations - data representation - data types, sign and magnitude, complements, fixed-point representation, floating-point representation, other binary codes, error detection codes.
**Unit II:** Basic computer organization – machine instructions – classification, function, addresses, size, addressing modes – instruction cycle - instruction sequencing. fundamental concepts – registers, register transfers, performing arithmetic or logic operations, memory read and write, execution of a complete instruction, branch instruction, Single bus, two bus, three bus organization, a complete processor - Control unit: - hardwired control, microprogrammed control, micro instructions-types.

**Unit III:** Arithmetic & Logic Unit - addition of positive numbers - fast adders - signed addition and subtraction - addition/subtraction logic unit - multiplication of positive numbers - array multiplier, sequential multiplier - signed number multiplication - multiplication using Booth's algorithm - fast multiplication - bit pair recording of multiplication, division-restoring and non restoring algorithms, floating point numbers and operations.


**References:**


S 1.6 Practical – Programming and Data Structures using C
Course No: 6

Prerequisites/ Exposure: None
Objective: To practically implement the techniques learned from course no 2 and 4.

Unit I: C Programming

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

Unit II: Data Structures and Algorithms

1. Implement stacks using arrays.
2. Implement queues, circular queue using arrays.
3. Implement sequential search and binary search techniques.
4. Implement linked lists and operations (add, insert, delete, search) on linked lists.
5. Implement stacks using linked list.
6. Implement queues using linked list.
7. Implement doubly linked lists.
8. Implement circular linked lists.
9. Implement binary tree and traversals
10. Implement Binary search trees and perform the operations on BST.
11. Implement various sorting algorithms.
12. Convert an infix expression to the postfix form using stacks.
13. Write a program to evaluate a postfix expression.
15. Implement Heap tree and operations.

Second Semester

S 2.1 Design and Analysis of Algorithms

Course No: 7

Objective:
- To introduce the concept of algorithmic approach for solving real-life problems.
- To teach basic principles and techniques of computational complexity.
- To familiarize with parallel algorithms and related techniques.


Unit II: Master’s theorem - solution to recurrence relations with full history - probabilistic analysis - linearity of expectations - worst and average case analysis of quick sort, merge sort, heapsort, binary search, hashing algorithms - lower bound proofs for the above problems - amortized analysis - aggregate, accounting and potential methods - analysis of Knuth-Morris-Pratt algorithm - amortized weight balanced trees.


Unit IV: Complexity - complexity classes - P, NP, Co-NP-Hard and NP-complete problems - Cook’s theorem - NP completeness reductions for clique, vertex cover, subset sum, Hamiltonian cycle and TSP.

References

8. Sahni, Data Structures, Algorithms and Applications in C++, Tata Mcgraw Hill.

S 2.2 Operating System Concepts
Course No: 8

Prerequisites/Exposure: S1.2 Advanced Data Structures, S 1.5 Computer Organization & Architecture.

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


**References:**

**S 2.4 Computer Networks**  
**Course No:** 9  
**L P C**  
4 0 4

**Prerequisites/ Exposure:** None

**Objectives:**
- To provide the student with a top down approach of networking starting from the application layer.
- To introduce computer networking in the back drop of Internet protocol stack.
- Be conversant with primitives of network application programming.


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


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

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

**References**
S 2.4 Computational Intelligence  
Course No: 9  

Prerequisites/ Exposure: None

Objectives: Introduce concepts of artificial intelligence and machine learning.

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

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

Unit III: Knowledge representation issues, representation and mappings, Representing simple facts in logic, Representing instances and ISA relationships, Computable functions and Predicates, Resolution, Natural deduction, Knowledge representation using rules, logic programming, forward versus backward reasoning, Symbolic reasoning under uncertainty- Nonmonotonic reasoning, Depth first search, Breadth first search.


Text Books:  
Proposed Syllabus for M.Sc. Computer Science - 2014 Admission onwards

References:

S 2.5a Computer Graphics (Elective I)

Course No: 11a

Prerequisites/ Exposure:

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

Unit I: Introduction - application and output devices for computer graphics - raster and random scan display, CRT, color CRT, flat panel, LCD, LED, DVST. Adapters - monochrome display adapter (MDA), CGA, hercules graphics card, enhanced graphics adapter, Professional graphics adapter, VGA, SVGA. Graphics software - GKS, PHIGS, OpenGL. Scan conversion - Points & lines, line drawing algorithms - DDA algorithm, Bresenham's line algorithm. Circle generation algorithm - Mid-point circle algorithm, Ellipse generation.


Unit III: Projection - 3D concepts & viewing pipeline, coordinate system, window to viewport coordinate transformation, parallel & perspective projection, projection matrix, view volume. 3D object representation - wireframe model, visible surface detection methods, depth comparison, Z-buffer algorithm, back face detection, BSP tree method, printer's algorithm, depth cueing.
Unit IV: Curves & Fractals - curve representation, surfaces, designs, spline representation, Bezier curves, cubic spline, beta spline, B-spline curves. Fractal's geometry, fractal generation procedure, classification of fractal, fractal dimension, fractal construction methods.


Reference:
5. OpenGL Redbook Version 1.1 (Online)

S2.5b Introduction to Soft Computing (Elective I)
Course No: 11b

Prerequisites/ Exposure:

Objectives:
- To give students knowledge of soft computing theories fundamentals.
- To expose the fundamentals of non-traditional technologies and approaches to solving hard real-world problems.

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


Unit III: Neural Model and Network Architectures, Perceptron Learning, Supervised
Hebbian Learning, Back-propagation, Associative Learning, Competitive Networks, Hopfield Network, Computing with Neural Nets and applications of Neural Network.

**Unit IV:** Introduction to Fuzzy Sets, Operations on Fuzzy sets, Fuzzy Relations, Fuzzy Measures, Applications of Fuzzy Set Theory to different branches of Science and Engineering.

**Unit V:** Advanced Topics - Support Vector Machines, Evolutionary computation (EC) - Evolutionary algorithms, Harmony search, Swarm intelligence.

**References:**


**S2.5c Web Technology (Elective I)**

**Course No:** 11c

<table>
<thead>
<tr>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

**Prerequisites/ Exposure:**

**Objectives:** Introduction tools for creating and maintaining websites - content development (HTML), client side scripting (JavaScript), web server (Apache), server side scripting (PHP), content management system (Joomla).

**Unit I:** Introduction to Web programming - Introduction to SGML features - HTML, XHTML, DHTML, XML - HTML Vs XML - Creating XML documents - Parsing an XML document - Writing well formed documents - Organizing elements with namespaces - Defining elements in a DTD - Declaring elements and attributes in a DTD. Overview of HTML - basic formatting tags - heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image. Attributes - align, color, bgcolor, font face, border, size. Navigation Links using anchor tag - internal, external, mail and image links. Lists - ordered, unordered and definition, Table tag, HTML Form controls - form, text, password, textarea, button, checkbox, radio button, select box, hidden controls,
Frameset and frames. CSS.

**Unit II:** Client side programming – Introduction – popular client side scripting languages - JavaScript - Introduction, Identifiers, Operators, Functions, Event handling, Classes, objects, Array, math, string, window object, Navigator DHTML Font, Text, Image change, Table expansion. JavaScript's object model; Strengths and weaknesses of JavaScript; Building and extending objects in JavaScript; Events in JavaScript; Event-handlers; Creating interactive forms; Introduction to cookies; using cookies in JavaScript & storing users choices in cookies. Encoding cookies; Browser objects: Object hierarchy, Creating Browser objects, Working with window, Document, History & location; Browser detection, Java to JavaScript communication.


**References :**
1. Thomas A. Powell, The Complete Reference HTML
2. E. Stephen Mack & Janan Platt, HTML 4.0 - No experience required.
6. H.M. Deitel, P.J. Deitel and A.B. Goldberg, Internet and World Wide Web: How to
Proposed Syllabus for M.Sc. Computer Science - 2014 Admission onwards

Program, 3rd edition, Pearson Education.
7. Wagner and R. Allen Wyke, Javascript, SAMS.

S2.5d Bio Informatics (Elective I)
Course No: 11d

Prerequisites/ Exposure:

Objectives:
• Expose students to the popular genomic and proteomic databases and to impart knowledge in processing and analyzing genomic data.
• Introduce advanced topics in bioinformatics.


Unit III: Sequence Alignment - Pair-wise sequence alignment, Need of scoring schemes - Penalizing gaps, Scoring matrices for amino acid sequence alignment, PAM Probability matrix and Log odds matrix, BLOSUM, Dot-plot visualization, Needleman-Wunsch algorithm- effect of scoring schemes - evaluate - BLAST and FASTA, Smith-Waterman algorithm for local alignment.

Unit V: Introduction to the Major Resources - NCBI, EBI and ExPASy - Nucleic acid sequence databases - GenBank, EMBL, DDBJ – Protein sequence databases - SWISS-PROT, TrEMBL, PIR_PSD - Genome Databases at NCBI, EBI, TIGR, SANGER - Procedures to access these databases and to make use of the tools available.

Text Books:
1. Mount D, Bioinformatics: Sequence & Genome Analysis, Cold spring Harbor press.

References:
3. Tao Jiang, Ying Xu and Michael Q. Zhang, Current Topics in Computational Molecular Biology, Ane Books.
S2.5e Computer Optimization Techniques (Elective I)
Course No: 11e

Prerequisites/ Exposure:

Objectives:
- To give an exposure for the student to the area of modeling techniques, numerical methods and algorithms.
- To realize the importance of various aspects of optimization techniques in industries like manufacturing and IT.
- To implement the knowledge of optimization techniques in real life problems.

Unit I: Linear Programming and Sensitivity Analysis - Two-variable LP-model, graphical and algebraic LP solutions, some LP applications, the Simplex Method and sensitivity analysis, primal-dual relationships and economic interpretation, dual simplex and generalized simplex algorithms and post-optimal analysis.

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

Unit III: Advanced Linear Programming and Applications - Simplex method fundamentals, revised simplex method and computational considerations, bounded variables algorithm, duality, parametric linear programming, goal programming formulations and algorithms.

Unit IV: Integer Linear Programming - Illustrative applications, integer programming algorithms, unimodularity and cutting-plane methods, traveling salesperson problem.

Unit V: Dynamic Programming 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
S2.5f Numerical and Statistical Methods (Elective I)
Course No: 11f

Prerequisites/Exposure:

Objectives:
- To provide the student with basic concepts in Statistics, Probability that can be applied for mathematical modeling of computer applications.


References:

1. E. Balagurusamy, Numerical Methods, 1999 Tata Mcgraw-Hill.
5. Mital Sethi, Linear Programming Pragathi Prakashan
S2.6 Practical 2 (S2.2 & S2.3)

Course No: 12

Prerequisites/ Exposure: S2.2 & S2.3

Objectives:
- To practically implement the theory parts covered in subjects S2.2 and S2.3

Unit I: Operating System

1. Shell programming: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).
2. Implement process creation using Process System Calls (fork(), wait(), exec(), stat(), readdir())
3. File System Calls (open(), read(), write())
4. Command simulation (ls, grep, cp, rm)
5. Process Scheduling (FCFS, SJF, Priority, Round robin) Interprocess Communication (Fibonacci & Prime nos, who | wc –l, Chat Messaging, Shared Memory, Producer-Consumer problem)
6. Memory Management (First Fit, Best Fit, FIFO Page Replacement, LRU Page Replacement)
7. File Allocation (Contiguous Allocation)

Unit II: Computer Networks

1. Design a LAN with a given set of requirements. The design should include topology, hardware and software requirements like cable, connectors, hubs/switches/bridges, interface cards along with a budget for the LAN. (Faculty in charge should give the requirements to the students).
2. Write a program to implement TCP Echo Client
3. Write a program to implement TCP Echo Server
4. Write a Program to check the Date and Time in TCP Date Time Client
5. Write a Program to check the Date and Time in TCP Date Time Server
6. Write a program to transfer a File using TCP.
7. Write a program to transfer Files using UDP.
8. Write a program to simulate the sliding window protocol.
9. Study of Network Simulators like NS2 / Glomosim
S2.7 Seminar
Course No: 13

The aim of this course is to introduce the student to research, and to acquaint him/her with the process of presenting his/her work through seminars and technical reports.

The student is expected to do an extensive literature survey and analysis in an area related to computer science, chosen by him/her, under the supervision of a faculty member from the department. The 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.

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

Third Semester

S3.1 Advanced Data Base Management System
Course No: 14

Prerequisites/ Exposure:

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

Unit I: Introduction: Purpose of Database Systems, Views of Data –Data Abstraction, Instances and Schemas, Data Independence ,Data Models – Hierarchical Data Model, Network Data Model, Relational Data Model, ER Data Model. Database Languages-DDL,

Unit II: Relational Database Design - Anomalies in a Database - Functional Dependency - Lossless Join and Dependency- Preserving Decomposition - normalization - normal forms - First, Second and Third Normal Form - Boyce Codd Normal Form - Multivalued, Dependency - Fourth Normal Form - Join Dependency - Project Join Normal Form - Domain Key Normal Form.

Unit III: Relational Database Query Languages - Basics of QBE and SQL. Data Definition in SQL - Data types, Creation, Insertion, Viewing, Updation, Deletion of tables, Modifying the structure of the tables, Renaming, Dropping of tables. Data Constraints - I/O constraints, Primary key, foreign key, unique key constraints, ALTER TABLE command - Database Manipulation in SQL - Computations done on table data - Select command, Logical operators, Range searching, Pattern matching, Grouping data from tables in SQL, GROUP BY, HAVING clauses, Joins - Joining multiple tables, Joining a table to itself. DELETE - UPDATE - Views - Creation, Renaming the column of a view, destroys view - Program with SQL - Data types: Using set and select commands, procedural flow, if, if /else, while, goto, global variables, Security - Locks, types of locks, levels of locks. Cursors - Working with cursors, Error Handling, Developing stored procedures, create, alter and drop, passing and returning data to stored procedures, using stored procedures within queries, building user defined functions, creating and calling a scalar function, implementing triggers, creating triggers, multiple trigger interaction.


Distributed Databases, Distributed Transactions, Commit Protocols for Distributed Databases.

References:

3. CJ Date, Introduction to Database Systems, Addison Wesley.

S3.2 Principles of Compilers
Course No: 15

Prerequisites/ Exposure:

Objectives:
- To introduce the concept of different phases of compiler.


Unit III: Intermediate code generation - DAG - Three Address Code - Addresses and Instructions - quadruples - triples - Static Simple-Assignment Form - Types and


References:
3. Tremblay, Sorenson, The Theory and Practice of Compiler Writing, BSP.
4. Torben Ægidius Mogensen, Basics of Compiler Design, Department Of Computer Science, University Of Copenhagen (Online Edition).

S 2.3 Object Oriented Programming Concepts using Java
Course No: 11

Unit I: 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 - The for Statement, The if Statement, The while and do while Statements, The switch Statement, The break
Statement, The continue Statement, Operators - Casts and Conversions, Arrays.


**Unit III**: Exceptions, Threads & IO in Java - The File and Standard Streams, Stream classes and interfaces, Using Byte Streams and Character Streams, Threads - Threads vs. Processes, Creating Threads, Runnable interface, Thread Class, Inter thread communication, Synchronization. Exceptions - Basic of java Exception Handling, Hierarchy, Developing user defined Exception Classes.

**Unit IV**: Applets, AWT & Swing - Applet class, Types of applet, skeleton, Applet tag, passing parameters, Event Handling, Delegation event model, Event classes, Listeners, AWT classes and window fundamentals, Frames, Working with fonts, graphics and colors, AWT controls, layouts and Menus, Dialogue Boxes. Swings - Japplets, icon, labels, Buttons, Textbox, combo box, Tables and Panes.

**Unit V**: Database and Sockets - JDBC - introduction, architecture, Drivers, connections, statements, resultset and Meta data. Sockets: Introduction to networking, InetAddress, url, socket, server sockets, Datagram.

Introduction to Unified Modeling Language, UML diagrams, Class diagrams, Object interaction diagrams, State and Activity diagrams, Component diagrams, Deployment diagrams. Introduction to Analysis Object Oriented System Analysis, Design and implementations.

**References:**

2. E Balaguruswamy, Programming in Java.
S3.4a Pattern Recognition (Elective II)
Course No: 17a
Prerequisites/Exposure:

<table>
<thead>
<tr>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

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

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

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

Unit III: Linear discriminant functions - linear discriminant functions and decision surfaces - generalized linear discriminant functions - 2-category linearly separable case - non-separable behaviour - linear programming algorithms, support vector machines - multilayer neural networks - feedforward operation and classification, backpropagation algorithm, error surface, backpropagation as feature mapping.

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

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

References:
2. Gonzalez R.C. & Thomson M.G., Syntactic Pattern Recognition - An Introduction,
Addison Wesley.
3. Fu K.S., Syntactic Pattern Recognition And Applications, Prentice Hall, Eaglewood cliffs

S3.4b Wireless and Mobile Networks (Elective II)
Course No: 17b
Prerequisites/ Exposure:
Objectives:
- To understand the fundamental concepts of wireless and mobile networks.
- To learn the basics of Wireless voice and data communications technologies.
- To build working knowledge on various telephone and satellite networks.
- To build skills in working with Wireless application Protocols to develop mobile content applications.
- To understand about the security aspects of Wireless Networks.
- To learn about Wireless Application Protocol and mobile operating systems.


Unit V: Java for Wireless Devices - Setting up the development environment - Basic Data types, Libraries (CLDC, MIDP) - UI Controls - Displayable and Display Image - Events and Event Handling - List and choice - Text box - Alerts - Persistent Storage - Record Stores - Records - Record Enumeration - Network MIDlets - The Connection Framework - Connection Interface - Making a connection using HTTP - Using datagram connection.
References:

S3.4c Cryptography & Network Security (Elective II)
Course No: 17c

Prerequisites/ Exposure:

Objectives: To be familiar with classical and modern encryption and decryption techniques and apply in the security system.


Unit V: Intruders - Intruders, Intrusion Detection, Password management. Malicious Software - Types, Viruses, Countermeasures, Worms, DDoS. Firewalls - Need -
Characteristics, Types, Firewall Basing, Location and Configuration – DMZ Networks, VPNs – Distributed Firewalls.

References:


S3.4d Advanced Web Technology (Elective II)
Course No: 17d
Prerequisites/ Exposure:

Objectives: To be familiar with the design and development process for distributed system.


Unit V: Python - SQLite integration - Features of SQLite, data types, Introduction to SQL commands - SELECT, DELETE, UPDATE, INSERT. Python functions for SQLite operations - database connection, database and table creation, selection, query, fetching results - Insertion and Deletion of data using Python - Displaying data from SQLite in webpage. Case study - Server MVC design pattern - Django/Plone (Choose any one of these).

References:

8. Peter Wentworth Jeffrey Elkner, Allen B. Downey, and Chris

S3.4e Virtualisation And Cloud Computing (Elective II)  
Course No:17e

Prerequisites/ Exposure:

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


**Unit II:** Virtualization - Basics of virtualization - Types of Virtualization - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Desktop virtualization - Server Virtualization - Linux KVM, Xen, Qemu, LXC, OpenVZ.

**Unit III:** Cloud Infrastructure - FOSS Cloud Software Environments - Eucalyptus, Opennebula, OpenStack – OpenStack Architecture – Compute, Object Storage, Image Service, Identity, Dashboard, Networking, Block Storage, Metering, Basic Cloud Orchestration and Service Definition.

**Unit IV:** Programming Model - Parallel and Distributed programming Paradigms – MapReduce, Twister and Iterative MapReduce – Mapping Applications - Programming Support – Apache Hadoop – HDFS, Hadoop I/O, Hadoop configuration, MapReduce on Hadoop.


**References:**
4. George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice (O'Reilly), O'Reilly
6. Katarina Stanoevska - Slabeva, Thomas Wozniak, Santi Ristol, Grid and Cloud
Computing – A Business Perspective on Technology and Applications”, Springer.

S3.4f Data Warehousing and Data Mining (Elective II)
Course No: 17f

Objectives:
• To provide the fundamentals on information retrieval and data mining techniques and focus on practical algorithms of textual document indexing, relevance ranking, web usage mining, text analytics, as well as their performance evaluations, that lays foundations for the Data Analytics.
• To give an exposure to the fundamentals of Data Analytics.

Unit I: Data Warehouse - Definition - Operational Database Systems Vs Data Warehouses - Multidimensional Model - From Tables and Spreadsheets to Data Cubes - Schemas for Multidimensional Databases - Measures - Concept Hierarchies - OLAP Operations in the Multidimensional Data Model - DataWarehouse Architecture.

Unit II: Data Mining - Introduction - Definition - Data Mining Functionalities - Major Issues in Data Mining - Data Preprocessing - Data Cleaning - Data Integration and Transformation - Data Reduction - Data Discretization and Concept Hierarchy Generation. Association Rule Mining - Efficient and Scalable Frequent Item set Mining Methods - Mining Various Kinds of Association Rules - Association Mining to Correlation Analysis - Constraint-Based Association Mining.

Unit III: Classification and Prediction - Issues Regarding Classification and Prediction - Classification by Decision Tree Introduction - Bayesian Classification - Rule Based Classification - Classification by Back propagation - Support Vector Machines - Associative Classification - Lazy Learners - Other Classification Methods - Prediction - Accuracy and Error Measures - Evaluating the Accuracy of a Classifier or Predictor - Ensemble Methods - Model Section.

Unit IV: Cluster Analysis - Types of Data in Cluster Analysis - A Categorization of Major Clustering Methods - Partitioning Methods - Hierarchical methods - Density-Based Methods - Grid-Based Methods - Model-Based Clustering Methods - Clustering High-Dimensional Data - Constraint-Based Cluster Analysis - Outlier Analysis.

Unit V: Graph Mining - Mining Object, Spatial, Multimedia, Text and Web Data - Multidimensional Analysis and Descriptive Mining of Complex Data Objects - Spatial Data Mining - Multimedia Data Mining - Text Mining - Mining the World Wide Web.
S3.5a Data Compression (Elective III)

Course No: 18a

Prerequisites/ Exposure:

4 0 4

Objectives:

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


Unit II: Dictionary methods - string compression, LZ77 sliding window, MZW, GIF images. Image Compression - Approaches to image compression, intuitive methods, image transform, test images, JPEG, Progressive image compression, Vector quantization.

Unit III: Wavelet Methods- Fourier transform, frequency domain, Fourier image compression, CWT and inverse CWT, Haar transform, filter bank, DWT, JPEG 2000. Video compression - analog video, Composite and component video, digital video, video compression, MPEG.

References:

S3.5b Pervasive Computing (Elective III)
Course No: 18b
Prerequisites/ Exposure:

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


Java 2 Enterprise Edition (Overview), Servlets, Enterprise Java Beans, Java Server Pages, Extensible Markup Language, Web Services, Model-View-Controller pattern.


**References:**

**S3.5c System Security (Elective III)**
**Course No:** 18c

**Objectives:**
An understanding of the differences between various forms of computer security, where they arise, and appropriate tools to achieve them.


**Unit II:** Program Security - Secure programs - Fixing Faults, Unexpected Behaviour, Types of Flaws. Non-malicious program errors - Buffer overflows, Incomplete Mediation.


**Unit IV:** Database Security - Security requirements - Integrity of Database, Confidentiality and Availability, Reliability and integrity, Sensitive data, Interface, Multilevel database, Proposals for multilevel security.


**References:**

**S4.1d Molecular Modeling and Simulation (Elective III)**
Course No: 18d

**Objectives:**
1. To understand application of simulation techniques to study molecular dynamics
2. To learn and apply the statistical approaches and models for phylogenetic analysis and tree reconstruction.

3. To understand the basis and nature of protein-protein interactions.

4. To understand principles of docking simulations.

**Unit I:** Overview of molecular modeling - Molecular modeling methods - Semi-empirical method and empirical method. Model Type - static, dynamic and probabilistic models. Models of growth and decay.

**Unit II:** System Modeling - Concept, Principles of Mathematical modeling, static physical model, stochastic activities, continuous and discrete simulation. Discrete system simulation - Probability concepts in simulation, random number generations and their testing, stochastic variable generation, Model Execution - Event driven versus Time driven.


**Unit IV:** Structural Modeling - Use of sequence patterns for protein structure prediction. Prediction of protein secondary structure from the amino acid sequences. Prediction of three dimensional protein structures. Protein structure classification - Two major classification schemes - CATH and SCOP. Protein structure prediction - Steps

**Unit V:** Molecular Visualization - Visualization of protein structure, Methods of studying proteins, Proteomics databases, Protein family databases, PDB file format. Software tools for 3D molecular graphic visualization - Rasmol - basic operations and steps in Rasmol to visualize the molecule, advantages of Rasmol, advantages of Swiss-PdbViewer. Docking Simulations - Rigid docking and Flexible docking.
Text Books:
1. Stephen Misener and Stephen A. Krawetz, Bioinformatics: Methods and
2. Gordan, Simulation and Modeling, PHI.
3. Tamar Schlick, Molecular Modeling and Simulation: An Interdisciplinary Guide,
   Springer, 2001
4. Narsingh Dev, System simulation and modeling, PHI
5. Leach, Andrew, Molecular Modelling: Principles and Applications, Prentice Hall.

References:
1. Sharma, Munjal and Shanker, A textbook of Bioinformatics, RASTOGI
   publications- New Delhi.
2. Des Higgins (Ed), Willie Taylor (Ed), Bioinformatics: Sequence, Structure and

S3.5e Fundamentals of Big Data (Elective III)
Course No: 18e
Prerequisites/Exposure:
4 0 4

Objectives:
- To cover the basics of big data
- To familiarize with big data technology and tools

Unit I: Introduction to Big Data - definition & importance of Big Data - four dimensions
   of Big Data - volume, velocity, variety, veracity - industry examples - terminologies -
   structured data, unstructured data, semi structured data, streaming data, real-time
   data, meta data, data at rest - relational databases and SQL - Non-Relational
   databases - big data sources that can change one's business - Integrating Big Data with
   traditional data - The role of the Data Scientist - Big Data Analytics in Industry
   Verticals.

Unit II: Key roles for a successful analytic project - Main phases of the lifecycle -
Developing core deliverables for stakeholders.

Unit III: Big Data analytics - Introduction - Concepts - Storing Big Data - Analyzing
your data characteristics - Selecting data sources for analysis - Eliminating redundant
data - Open source technology for Big Data analytics - Predictive analytics -
Crowdsourcing analytics - Computing platforms, limitations, and emerging
technologies - Consumption of analytics - Modern analytic approaches - ensemble
modeling, commodity models, and text analysis.

**Unit IV:** Introduction to MapReduce/Hadoop for analyzing unstructured data - design patterns – Filtering Patterns - Join Patterns - Meta Patterns - Hadoop ecosystem of tools - In database Analytics - MADlib and Advanced SQL Techniques, NoSQL, JSON store, MDX.

**Unit V:** Introduction to learning and knowledge analytics - Rise of Big Data - Big Data From Technology Perspective - Hadoop - Components of Hadoop, Application Development in Hadoop, The Distributed File System - HDFS, GPFS, Hadoop Cluster Architecture, Batch Processing - Low Latency NoSQL.

**References:**
1. Michael Minelli, Michele Chambers and, Ambiga Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses.
3. Tom White, Hadoop, The definitive guide, O'Reilly Media, 2010
5. Faraz Rabbani, Ali Roghani, Big Data Analytics For Beginners.
6. Alex Holmes, Hadoop in practice, Manning Publications, 2012
7. Donald Miner, Map Reduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems, O'Reilly Media, 2012
11. Bill Franks, Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics
12. Thomas H. Davenport, Big Data at Work: Dispelling the Myths, Uncovering the Opportunities

**S3.5f Web Engineering (Elective III)**

**Course No:** 18f

**Prerequisites/ Exposure:**

**Objectives:**
- To understand the concepts, principles, strategies, and methodologies of web applications development.
To understand and apply web development processes.

**Unit I:** Web Engineering (WE) - Introduction - Motivation - Categories & Characteristics of Web Applications - Product related, usage related and development related - Evolution of WE.


**Text Books:**

**References:**

**S3.6 Practical 3 (S3.1 & S3.3)**

<table>
<thead>
<tr>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**Unit I : Advanced DBMS**

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. Implementing use of triggers, cursors & procedures.

**Unit II : Java**

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

**S4.1a Digital Image Processing (Elective IV)**

Course No: 19a

Prerequisites/Exposure:

**Objective:** To be familiar with processing of the images, recognition of the pattern and their applications.

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

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

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

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

References:

2. B. Chanda and D.D. Majumder, Digital Image Processing and Analysis, PHI

S4.1b Advanced Topics in Database Design (Elective IV)

Course No: 19b

Prerequisites/ Exposure:

Objective:
- To study the advanced database techniques beyond the fundamental database techniques.

Unit I: 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.

Unit II: 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.

Unit III: 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, ORDBMS.

Unit IV: 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.

**Unit V:** Object Databases on the Web and SemiStructured 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, Fundamentals of Database Systems [4e], Pearson Education
5. C.J.Date, Longman, Introduction to Database Systems, Pearson Education

**S4.1c Software Development for Portable Devices (Elective IV)**

**Course No:** 19f

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

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


**Unit III:** Introduction to android and smart phones, Android Architecture & Virtual Machine, Mobile Technology terminologies, setting up the environment, Setting up Emulators, android fundamentals - Activities and Applications Activity Life Cycles Activity Stacks, Activity States, introduction to manifest, resources & R.java , assets, Values - strings.xml - Form widgets, views, Layouts & Drawable Resources - XML Layouts, Linear Layouts, Relative layouts, Table Layouts, android Widgets, UI XML Specifications Events, Bundles & Intents- Explicit Intents Implicit Intents Event Broadcasting with Intents Event Reception with Broadcast Receivers, Adapters and Data Binding.

**Unit IV:** Files, Content Providers, and Databases - Saving and Loading Files, SQLite Databases - Android Database Design - Exposing Access to a Data Source through a Content Provider Content Provider Registration Native Content Providers, Android Debug Bridge(adb) tool, Linkify.

**Unit V:** Adapters and Widgets , Notifications , Custom components Threads running on UI thread, Worker thread Handlers & Runnable AsynTask(in detail), Playing Audio and Video, Recording Audio and Video, Using the Camera to Take and Process Pictures. Networking & Location based services - Live Folders, Using sdcards – Reading and writing, XML Parsing - JSON Parsing - Including external libraries in applications, Map-Based Activities, Maps via intent and Map Activity GPS, Location based Services configuration, Geocoding, Accessing Phone services(Call,SMS,MMS) Network connectivity services, Using Wifi & Bluetooth Action bar tabs and custom views on Action bars. Introduction to cross platform application development - ruby on rail, phone gap (notions only).
References:
1. Html5 Black Book: CoversCss3, Javascript, Xml, Xhtml, Ajax, Php And J query, Kogent Learning Solutions Inc.
2. KESSLER, Programming HTML 5 Applications, OReilly
3. Android wireless application development, second edition by shane conder, Lauren darcey – Addison – Welsey
4. Android Application Development by Rick rogers, John Lombardo – O’Reilly
5. Professional Android 2 application development by Reto Meier - Wrox

S4.1d Storage Area Networks

Course No: 19h


Unit V: Designing and building a SAN- Design considerations Business requirements Physical layout Placement Storage pooling Data availability Connectivity scalability migration manageability fault tolerance and resilience - prevention of congestion routability- backup and restoration - SAN Security & iSCSI Technology Basic security guidelines implementing SAN security Backup and restoration iSCSI technology - Future of SANS.
References:
6. Christopher Poelker, Storage Area Networks for Dummies.

S4.1e Semantic Web
Course No: 19e

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


References:


S4.2f Advanced Java Programming

Course No: 19f

Prerequisite:

Objectives: To learn the advanced features of Java programming language.

Unit I: RMI & Servlets Introduction, Architecture, defining remote objects, creating stubs and skeletons, serializable classes, Accessing remote objects, factory classes, dynamically loaded classes, RMI activation, registering remote objects. Servlets, generic servlet, servlets that access request headers, Develop servlets that manipulate response headers, HTTP servlets, Forms, HTTP Proctols - Configuring Tomcat Server, Servlet Context, servlet context listener, Servelet Chaining.

Unit II: JNDI & EJB - Architecture, context initial context class, objects in a context, binding objects, accessing directory services, attributes and attribute interface modifying directory entities, creating directories entities. EJB roles, architecture, container, implementing a basic EJB object, Implementing session beans, implementing Entity bean, Deploying an enterprise bean object.

Unit III: Java Server Pages: 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 Describe how to build Web page layouts from reusable 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.


References:
2. Java Beans Developers Resource – PHI
4. Madhusudhan Konda, Just Hibernate, Oreilly

20 S4.5 Major Project
Course No: 20

Major project work is to be done individually by each student, under the guidance of a faculty member of the concerned department. Exposure to Software Engineering (SE) Principles and an insight to the Research Methodology (RM) is to be imparted to the student so that (s)he can proceed with the project work as per the underlying principles of SE/RM.

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

There shall be an evaluation committee (EC) for the internal evaluation of the work. EC should consists of HOD, at least two senior most faculty members and the guide of the student. EC can set a schedule for the evaluation of the work in different stages. For eg, a software development project can be evaluated in 5 stages – problem formulation, analysis, design, implementation and testing.

At the time of external evaluation, if the performance is below the mark, student will be given a chance to reappear within 3 months to present the work again, after incorporating the changes suggested by the external examiner. A certificate stating that the changes suggested by the external examiner are incorporated in the revised report.
is to be attached with the revised report which is signed by the student, guide and the 
HOD.

**Evaluation (Internal)**
The internal evaluation will be done by the EC in periodic intervals. 5% weightage is to 
be given for the evaluation of the SE/RM course work and remaining 45% weightage is 
for the work.

**Evaluation (External)**

**Guidelines for submission of report**

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

It is a test of the student’s ability to undertake and complete a sustained piece of 
independent research and analysis / application development, and to write up the work 
in a coherent form according to the rules and conventions of the academic community.

A satisfactory dissertation should not only be adequate in its methodology, in its 
analysis and in its argument, and adequately demonstrate its author’s familiarity with 
the relevant literature; it should also be written in correct, coherent language, in an 
appropriate style, correctly following the conventions of citation. It should, moreover, 
have a logical and visible structure and development that should at all times assist the 
reader’s understanding of the argument being presented and not obscure it. The layout 
and physical appearance of the dissertation should also conform to university 
standards.

The dissertation is to be prepared in tex format (either Latex or using an equivalent 
Windows tex variant such as MikeTex). The format of the report will be distributed 
shortly.

**Syllabus for Fundamentals of Software Engineering & Research Methodology**

Note:-
- This course is to be offered in the last semester for 20 hours. An evaluation test 
is to be conducted at the end of the session.
Unit I (4 Hours)

Unit II (4 Hours)

Unit III (3 Hours)

Unit IV (2 Hours)

Unit V (2 Hours)

Unit VI (2 Hours)

Unit VI (4 Hours)
How to perform a literature review - Sampling Methods - Data Processing and Analysis strategies - Data Analysis with Statistical Packages - Hypothesis-testing - Generalization and Interpretation.

Unit VII (3 Hours)
Research Reports - Structure and Components of Research Report, Types of Report, Layout of Research Report, Mechanism of writing a research report
References: