VIMALA COLLEGE (AUTONOMOUS)

(NAAC Re-accredited (3rd Cycle): A Grade, CGPA-3.50)

(Affiliated to University of Calicut)

POST GRADUATE DEGREE OF

MASTER OF SCIENCE (M.Sc.)

IN

COMPUTER SCIENCE
(Specialized in Data Science)

(CHOICE BASED CREDIT AND SEMESTER SYSTEM)

UNDER THE

FACULTY OF COMPUTER SCIENCE

SYLLABUS

(FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2020 – 21 ONWARDS)

VIMALA COLLEGE
ENGINEERING COLLEGE P O, THRISSUR
KERALA-680009
INDIA

CONTENTS

REGULATIONS FOR THE DEGREE OF

MASTER OF SCIENCE

COMPUTER SCIENCE

(Specialized in Data Science)

PROGRAMME OBJECTIVES

The course of the MSc (Computer Science) programme is designed with the following objectives:

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

PROGRAMME SPECIFIC OBJECTIVES

The course of the M.Sc. Computer Science (Specialized in Data Science) program is designed with the following objectives:

1. To train and equip the students to meet the requirements of the Software industry in the country and outside
2. Analyse a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions
3. To equip students to combine computer and statistical sciences to develop top edge tools to efficiently deal with data processing
4. Apply principles of Data Science to the analysis of business and real time problems
5. Apply algorithms to build machine intelligence in various contexts of computing
6. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles

PROGRAMME STRUCTURE

1. The programme includes three types of courses, viz., Core courses (Code C), Elective Courses (Code E) and Audit Courses (Code A).

2. Every student of the MSc Computer Science programme shall have to work on a project/dissertation of not less than 8 credits under the supervision of a faculty member as per the curriculum. Project/dissertation shall be treated as Core Courses. Project Work is mandatory for all regular programmes and Comprehensive Viva-voce is optional and these shall be done in the end semester. The combined Credit for the Project Work and Comprehensive Viva-voce shall not be more than 8 (eight) credits subject to a minimum of 4 (four) credit for Project Work. All students have to submit a Project Report/Dissertation in the prescribed structure and format as a part of the Project Work undertaken

3. Total credit for the programme shall be 80 (eighty), this describes the weightage of the course concerned and the pattern of distribution is as detailed below

a. Total Credit for Core Courses shall not be less than 60 (sixty).

b. Total Credit for Elective Course shall not be less than 12 (twelve) and not more than 20 (Twenty).

c. Total Credits for Comprehensive Viva-voce and Project Work combined together shall be 8 (eight) subject to a minimum of 4 (four) credit for Project Work.

d. Total credit in each semester shall vary between 18 to 22.

e. No course shall have less than 2 credits and more than 5 credits.

4. Elective courses shall be spread over either in the Third & Fourth Semesters combined.

5. Audit Courses: There will be two Audit Courses (Ability Enhancement Course & Professional Competency Course) with 4 credits each. These have to be done one each in the first two semesters. The credits will not be counted for evaluating the overall SGPA & CGPA. The colleges shall conduct examination for these courses. Students have to obtain only minimum pass requirements in the Audit Courses.

6. A student shall accumulate a minimum of 80 credits for the successful completion of the programmes

ATTENDANCE

1. The students admitted in the PG programmes in affiliated colleges shall be required to attend at least 75 percent of the total number of classes (theory/practical) held during each semester. The students having less than prescribed percentage of attendance shall not be allowed to appear for the Semester examination.
2. Condonation of shortage of attendance for a maximum of 9 days (10% of the working days in a semester) in the case of single condonation and 18 days (20% of the working days in a semester) in the case of double condonation in a semester subject to a maximum of two times (for single condonation only) during the whole period of Post Graduate programme may be granted by the Principal as per the existing procedures. In the case of double condonation, only one condonation shall be allowed during the entire programme.
3. Benefit of condonation of attendance will be granted to the students on health grounds, for participating in University Union activities, meeting of the University bodies / Govt. bodies and participation in other extracurricular activities on production of genuine supporting documents, with the recommendation of the Head of the Department concerned.
4. A student who is not eligible for such condonation shall be observed and the principal should intimate the details of these candidates at the commencement of the next semester.
5. Women students can avail maternity leave as per the existing university rules.

EXAMINATION

1. There shall be End Semester examination at the end of each semester.
2. Practical examinations shall be conducted by the College at the end of each semester. There will be one internal and one external examiner for the conduct of End Semester Practical examination.
3. Project Work / Dissertation shall be evaluated at the end of the programme only. There shall be both Internal and External evaluation for the Project Work.

4. There shall be one end-semester examination of 3 hours duration for each theory course and practical course.

EVALUATION AND GRADING

1. Evaluation: The evaluation scheme for each course shall contain two parts; (a) Internal / Continuous Assessment (CA) and (b) External / End Semester Evaluation (ESE).
2. Of the total, 20% weightage shall be given to Internal evaluation / Continuous assessment and the remaining 80% to External/ESE and the ratio and weightage between Internal and External is 1:4.
3. Primary evaluation for Internal and External shall be based on 6 letter grades (A+, A, B, C, D and E) with numerical values (Grade Points) of 5, 4, 3, 2, 1 & 0 respectively.
4. Grade Point Average: Internal and External components are separately graded and the combined grade point with weightage 1 for Internal and 4 for external shall be applied to calculate the Grade Point Average (GPA) of each course. Letter grade shall be assigned to each course based on the categorization based on Ten point Scale.
5. Evaluation of Audit Courses: The examination and evaluation shall be conducted by the college itself either in the normal structure or MCQ model from the Question Bank and other guidelines provided by the College/BoS. The Question paper shall be for minimum 20 weightage and a minimum of 2 hour duration for the examination.

INTERNAL EVALUATION – CONTINUOUS ASSESSMENT

1. This assessment shall be based on a predetermined transparent system involving periodic written tests, assignments, seminars and viva-voce in respect of theory courses and based on tests, lab skill and records/viva in respect of practical courses.
2. The criteria and percentage of weightage assigned to various components for internal evaluation are as follows :
   a. Theory: The weightage assigned to various components for internal evaluation for theory papers is as shown below.

Sl.No	Component	Percentage	Weightage
1	Examination /Test	40%	2
2	Seminars / Presentation	20%	1
3	Assignment	20%	1
4	Attendance	20%	1

To ensure transparency of the evaluation process, the internal assessment grade awarded to the students in each course in a semester shall be published on the notice board at least one week before the commencement of external examination. There shall not be any chance for improvement for internal grade.

The course teacher shall maintain the academic record of each student registered for the course, which shall be forwarded to the Controller of Examination.

b. Practical: The mark distribution to award internal continuous assessment marks for practical course should be as follows:

Sl.No	Component	Percentage	Weightage
1	Lab Skill	40%	4
2	Records/viva	30%	3
3	Practical Test	30%	3

Note:

- All students should have a rough record (observation note book) in which they write all the works to be carried out in the lab prior to his/her entering the lab. (S)he may also note down the i/p and o/p that (s)he gives for program verification in the observation note book (rough record).
- All lab works should be neatly recorded in a Laboratory Record Book (Fair Record) in written form. However program results can be pasted in the left hand side of the fare record.
- Chairperson, Board of Examination (PG) has to prepare the modalities of the practical papers (list of experiments to be done, number of minimum experiments required in the practical record, etc) and distributed to all departments concerned, at the beginning of each semester itself. Model lists of experiments are provided with the syllabus for each practical session.
- No candidate will be permitted to attend the end-semester test unless he/she produces certified record of the laboratory.

1. Grades shall be given for the internal evaluation are based on the grades A+, A, B, C, D & E with grade points 5,4,3,2, 1 & 0 respectively. The overall grades shall be as per the Ten Point scale.
2. There shall be no separate minimum Grade Point for internal evaluation.
3. To ensure transparency of the evaluation process, the internal assessment marks awarded to the students in each course in a semester shall be published on the notice board before 5 days of commencement of external examination.
4. There shall not be any chance for improvement of internal marks.
5. The course teacher shall maintain the academic record of each student registered for the course, which shall be forwarded to the Controller of Examination, after being endorsed by the Head of the Department.
6. For each course there shall be class test/s during a semester. Grades should be displayed on the notice board. Valued answer scripts shall be made available to the students for perusal.
7. Each student shall be required to do assignment/s for each course. Assignments after valuation must be returned to the students. The teacher shall define the expected quality of the above in terms of structure, content, presentation etc. and inform the same to the students. Punctuality in submission is to be considered.
8. Every student shall deliver Seminar / Presentation as an internal component for every course and must be evaluated by the respective course teacher in terms of structure, content, presentation and interaction. The soft and hard copies of the seminar report are to be submitted to the course teacher.
9. All the records of Continuous Assessment (CA) must be kept in the college and must be made available for verification by university, if asked for.

Calculation of overall internal grade for one theory course will be done as shown below:

Components	**Weightage (W)**	**Grade Awarded**	**Grade Point(GP)**	**Weightage GP**	**Overall Gradeof the course**
Examination /Test	4	A	4	16	Weighted GP/Total Weight 43/10 = 4.30
Seminars / Presentation	3	A+	5	15	
Assignments	3	A	4	12	
Total	10			43	O

Calculation of overall internal grade for one Lab Course will be done as shown below:

Components	Weightage (W)	Grade Awarded	Grade Point(GP)	Weightage GP	Overall Gradeof the course
Lab Skill	2	A	4	8	Weighted GP/Total Weight 22/5 = 4.40
Records/viva	1	A+	5	5	
Practical Test	1	A	4	4	
Viva-voce	1	A+	5	5	
Total	5			22	O

EXTERNAL / END SEMESTER EVALUATION (ESE)

1. Thesemester-endexaminationsintheorycoursesshallbeconductedbythe College.Theevaluation oftheanswerscriptsshall be done by examiners based on a well-defined scheme of valuation.
2. After the external evaluation, only Grades are to be entered in the space provided in the answer script for individual questions and calculations need to be done only up to the Cumulative Grade Point (CGP) and all other calculations including grades are to be done by the College.
3. Students shall have the right to apply for revaluation or scrutiny as per rules within the time permitted for it. Photocopies of the answer scripts of the external examination shall be made available to the students for scrutiny on request by them as perrules.
4. Theexternalevaluationshallbedoneimmediatelyafterthe examination.
5. Thelanguageofwritingtheexaminationshallbe English.
6. Pattern of questions for external/ESE (theory courses):

a. Questionsshallbesettoassesstheknowledgeacquired,standard,andapplication of knowledge, application of knowledge in new situations, critical evaluation of knowledge and the ability to synthesize knowledge. Due weightage shall be given to each module based on content/teaching hours allotted to each module.

b. It has to be ensured that questions covering all skills are set. The setter shall also submit a detailed scheme of evaluation along with the question paper.

c. A question paper shall be a judicious mix of short answer type, short essay type/problem solving type and long essay type questions.

d. The question shall be prepared in such a way that the answers can be awarded A+, A, B, C, D, E Grades.

e. Weightage: Different types of questions shall be given different weightages to quantify their range given in the following model:

Sl. No.	Type of Questions	Individual weightage	Total Weightage	Number of questions to be answered
1	Short Answer type questions	2	2×4 = 8	4 out of 7
2	Short essay/ problem solving type	3	3×4 = 12	4 out of 7
3	Long Essay type questions	5	5×2 = 10	2 out of 4
Total			30	18

f. Questions should be asked as far as possible from all modules following a uniform distribution.

A sample ESE evaluation sheet of a theory course is illustrated below:

Type of Question	Qn. No	Grade Awarded	Grade Point	Weightage	Weighted Grade Point	Calculation
Short Answer type	1	A+	5	2	10	Overall Grade of the theory paper = Sum of Weighted Grade Points /Sum of the weightage 115/30 = 3.83 = Grade A+
	2	-	-	-	-	
	3	A	4	2	8	
	4	C	2	2	4	
	5	-	-	-	-	
	6	A	4	2	8	
	7	-	-	-		
Medium Essay type	8	B	3	3	9	
	9	A+	5	3	15	
	10	-	-	-	-	
	11	-	-	-	-	
	12	-	-	-	-	
	13	A	4	3	12	
	14	B	3	3	9	
Long Essay type	20	A+	5	5	25	
	21	-	-	-	-	
	22	-	-	-	-	
	23	B	3	5	15	
	24	-	-	-	-	
TOTAL				30	115	

g. End Semester Evaluation in Practical Courses shall be conducted and evaluated by both Internal and External Examiners.

Mark distribution for practical courses shall be as follows:

Component	Weightage
Algorithm/Flow diagram/UI diagram/Class	6
Implementation	6
Result/ Output	6
Record	6
Viva	6
Total	**30**

h. A sample ESE evaluation sheet of a theory course is illustrated below:

Type of Question	Grade Awarded	Grade Point	Weightage	Weighted Grade Point	Calculation
Algorithm/Flow diagram/UI diagram/Class Diagram	A	4	6	24	114/30 =3.80
Implementation	A	4	6	24	
Result/ Output	B	3	6	18	
Record	A	4	6	24	
Viva	A	4	6	24	
Total			**30**	**114**	**O**

EVALUATION OF PROJECT WORK / DISSERTATION

1. There shall be External and Internal evaluation for Project Work done and the grading system shall be followed.
2. One component among the Project Work evaluation criteria shall be Viva-voce (Project Work related) and the respective weightage shall be 40%.
3. Consolidated Grade for Project Work is calculated by combining both the External and Internal in the Ratio of 4:1 (80% & 20%).
4. For a pass in Project Work, a student has to secure a minimum of P Grade in External and Internal examination combined. If the students could not secure minimum P Grade in the Project work, they will be treated as failed in that attempt and the students may be allowed to rework and resubmit the same in accordance with the University exam stipulations. There shall be no improvement chance for Project Work.
5. The External and Internal evaluation of the Project Work shall be done based on the following criteria and weightages as detailed below:

Sl. No	Criteria	% of Weightage	Weightage	
			External	Internal
1	Relevance of the topic and Statement of problem, Methodology & Analysis Quality of Report & Presentation	60%	24	6
2	Viva-voce	40%	16	4
Total Weightage		**100%**	**40**	**10**

The first component for 60% weightage can be sub-divided into following project implementation components:

Sl No.	Components	Weightage	
		External	Internal
1	Relevance of the Topic, Statement of Objectives, Methodology	2	2
2	Quality of Literature Survey/Product Review	2	
3	Quality of Analysis Phase	2	
4	Quality of Design Phase	2	
5	Quality of Implementation/Simulation	4	2
6	Quality of Testing/Result Analysis	2	
7	Quality of Contributions	2	
8	Identification of Future Work	1	2
9	Quality of Project Report	4	
10	Publications/Presentations out of the Project Work*	1	
11	Quality of Presentation	1	
12	Demonstration of the Project Work	1	
13	General Viva Voce	16	4
	Total	**40**	**10**

DIRECT GRADING SYSTEM

1. Direct Grading System based on a 10 – Point scale I s used to evaluate the performance (External and Internal Examination of students)
2. For all courses (Theory & Practical)/Semester/Overall Programme, Letter grades and GPA/SGPA/CGPA are given on the following way:

Grade	Grade Points
A+	5
A	4
B	3
C	2
D	1
E	0

a. First Stage Evaluation for both Internal and External done by the Teachers concerned in the following Scale :

b. The Grade Range for both Internal & External shall be:

Letter Grade	Grade Range	Range of Percentage (%)	Merit Indicator
O	4.25 - 5.00	85.00 - 100.00	Outstanding
A+	3.75 - 4.24	75.00 - 84.99	Excellent
A	3.25 - 3.74	65.00 - 74.99	Very Good
B+	2.75 - 3.24	55.00 - 64.99	Good
B	2.50 - 2.74	50.00 - 54.99	Above Average
C	2.25 - 2.49	45.00 - 49.99	Average
P	2.00 -2.24	40.00 - 44.99	Pass
F	< 2.00	Below 40	Fail
I	0	-	Incomplete
Ab	0	-	Absent

1. No separate minimum is required for Internal evaluation for a pass, but a minimum P Grade is required for a pass in the external evaluation. However, a minimum P grade is required for pass in a course.
2. A student who fails to secure a minimum grade for a pass in a course will be permitted to write the examination along with the next batch.
3. Improvement of Course- The candidates who wish to improve the grade / grade point of the external examination of a course/s they have passed already can do the same by appearing in the external examination of the concerned semester along with the immediate junior batch
4. Betterment Programme One time- A candidate will be permitted to improve the CGPA of the Programme within a continuous period of four semesters immediately following the completion of the programme allowing only once for a particular semester. The CGPA for the betterment appearance will be computed based on the SGPA secured in the original or betterment appearance of each semester whichever is higher.

SEMESTER GRADE POINT AVERAGE (SGPA) - CALCULATION

The SGPA is the ratio of sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses taken by a student.

After the successful completion of a semester, Semester Grade Point Average (SGPA) of a student in that semester is calculated using the formula given below.

Semester Grade Point Average - SGPA (Sj) = Σ(Ci x Gi) / Cr

(SGPA= Total Credit Points awarded in a semester / Total credits of the semester)

Where 'Sj' is the j^{th} semester , 'Gi' is the grade point scored by the student in the i^{th} course 'Ci ' is the credit of the i^{th} course, 'Cr' is the total credits of the semester.

CUMULATIVE GRADE POINT AVERAGE (CGPA) CALCULATION

Cumulative Grade Point Average (CGPA) = **Σ** (Ci x Si) / Cr

(CGPA= Total Credit points awarded in all semesters/Total credits of the programme)

Where C_1 is the credit of the I^{st} semester S1 is the SGPA of the 1^{st} semester and Cr is the total number of credits in the programme. The CGPA is also calculated in the same manner taking into account all the courses undergone by a student over all the semesters of a programme. The SGPA and CGPA shall be rounded off to 2 decimal points.

For the successful completion of a semester, a student should pass all courses and score a minimum SGPA of 2.0. However, the students are permitted to move to the next semester irrespective of their SGPA.

MSc. COMPUTER SCIENCE (Specialized in Data Science) PROGRAMME STRUCTURE

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

SEMESTER I

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
1.1	CSS1C01	Discrete Mathematical Structures	4	1	4	5	4	0	4
1.2	CSS1C02	Advanced Data Structures	4	1	4	5	3	2	5
1.3	CSS1C03	Theory of Computation	4	1	4	5	4	0	4
1.4	CSS1C04	The Art of Programming Methodology	4	1	4	5	2	2	4
1.5	CSS1C05	Computer Organization Architecture	4	1	4	5	4	0	4
1.6	CSS1L01	Practical I	2	1	4	5	0	4	4
1.7	CSS1A01	Introduction to Research (Ability Enhancement Audit Course)	4	5	0	5	0	0	0
Total Credits (Excluding Audit Course): 22							**17**	**8**	**25**

SEMESTER II

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
2.1	CSS2C06	Design and Analysis of Algorithms	4	1	4	5	4	0	4
2.2	CSS2C07	Operating System Concepts	4	1	4	5	3	2	5
2.3	CSS2C08	Computer Networks	4	1	4	5	4	0	4
2.4	CSS2C09	Computational Intelligence	4	1	4	5	4	0	4
2.5	CSS2C10	Principles of Software Engineering	4	1	4	5	4	0	4
2.6	CSS2L02	Practical II	2	1	4	5	0	4	4
2.7	CSS2A02	Term Paper (Professional Competency Audit Course)	4	5	0	5	0	0	0
Total Credits (Excluding Audit Course): 22							**17**	**8**	**25**

SEMESTER III

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
3.1	CSS3C11	Advanced Database Management System	4	1	4	5	3	1	4
3.2	CSS3C12	Object Oriented Programming Concepts	4	1	4	5	2	3	5
3.3	CSS3C13	Principles of Compilers	4	1	4	5	4	0	4
3.4	CSS3E01	Elective I	4	1	4	5	4	0	4
3.5	CSS3E02	Elective 2	4	1	4	5	4	0	4
3.6	CSS3L03	Practical III	2	1	4	5	0	4	4
Total Credits (Excluding Audit Course): 22							**17**	**8**	**25**

List of Elective Courses for CSS3E01	
Course Code	Course Name
CSS3E01a	Basics of Data Mining & Feature Engineering
CSS3E01b	Introduction to Data warehouse
CSS3E01c	Fundamentals of Big Data
CSS3E01d	Statistical Techniques for Data Science
CSS3E01e	Computing Approaches

List of Elective Courses for CSS3E02	
Course Code	Course Name
CSS3E02a	Machine Learning
CSS3E02b	Pattern Recognition
CSS3E02c	Predictive Analytics and Data Modelling
CSS3E02d	Virtualization and Cloud Computing
CSS3E02e	Numerical and Statistical Methods

SEMESTER IV

No	Course Code	Course Name	C	Weightage			Hrs/Week		
				I	E	T	L	P	T
4.1	CSS4E03	Elective 3	3	1	4	5	5	0	5
4.2	CSS4E04	Elective 4	3	1	4	5	5	0	5
4.3	CSS4P01	Project Requirements Analysis & Design Related Discussion	8	1	4	5	3	1	4
		Project Coding, Testing & Implementation Related Discussion					2	2	4
		Project Evaluation & Assessment					2	0	2
		Project Lab Work					0	5	5
Total Credits (Excluding Audit Course): 14							**17**	**8**	**25**

List of Elective Courses for CSS4E03	
Course Code	Course Name
CSS4E03a	Deep Learning
CSS4E03b	Advanced Computing Technologies in Data Science
CSS4E03c	Natural Language Processing
CSS4E03d	Computer Optimization Techniques
CSS4E03e	Soft Computing

List of Elective Courses for CSS4E04	
Course Code	Course Name
CSS4E04a	Text Analytics
CSS4E04b	IoT Analytics
CSS4E04c	Image And Video Analytics
CSS4E04d	Visual Data Analytics
CSS4E04e	Healthcare Data Analytics

SEMESTER I

CSS1C01 –DISCRETE M ATHEMATICAL STRUCTURES

Objective

- To introduce discrete mathematics concepts necessary to understand basic foundation of Computer Science.
- Understand the mathematics behind Computer Science
- Understand the applications of graphs, sets, groups in Computer Science

Course Outcome

- Understand the concepts of Set Theory and its application
- Apply the concepts of predicate & proportional calculus
- Understand and apply the concepts of Functions and relations
- Understand and apply the various concepts of group theory
- Study various concepts of Graph theory and Tree structures
- Apply the concepts of various graph and tree algorithms

Course Outline

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

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

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

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

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

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 Publications Pvt. Ltd, ISBN: 9780574217554.
3. J. K. Sharma, Discrete Mathematics, Macmillan Publishers India Limited, ISBN: 1403924759.
4. J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Application to Computer Science, McGraw-Hill Companies, ASIN: B001FPXR5Y.

CSS1C02 – ADVANCED DATA STRUCTURES

Objective

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

Course Outcome

- Understand the concepts of data structures and Algorithms.
- Mathematical tools to analyze the performance of data structures
- Implement various linear and non-linear data structures such as arrays, queues, stacks, linked list, trees, graphs etc.
- Apply the concepts graphs, trees and related algorithms.
- Implement various hashing techniques.

Course Outline

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

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

Unit III: Non-linear Data Structures - trees - terminology - tree traversals algorithms - Binary trees -

threaded binary trees - binary search trees - traversals and operations on BST heap Tree - balanced trees - M-way trees - B and B+ trees, Red Black Tree, Digital Search Tree, Tries, Treaps, Huffman algorithm for extended binary tree - operations and their implementation. Graphs - representation of graphs – operations - traversals and their implementation.

Unit IV: Hashing - overview of hashing - hash tables - hash functions and their computations open addressing - linear probing - quadratic probing - double hashing algorithms and their implementations - rehashing - extendable hashing - separate chaining - hashing efficiency - heaps - overview of heaps - implementation and operations.

Unit V: Heap structures - Min-Max heaps - Heaps - leftist heaps - binomial heaps - Fibonacci heaps - binary heaps - skew heaps - pairing heaps - applications - amortized analysis an unrelated puzzle - Binomial queues - skew heaps - Fibonacci heaps - Splay trees.

References:

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley, ISBN: 978-0201000238.
2. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, Silicon Press, ISBN: 0929306406.
3. Richard F. Gilberg and Behrouz A. Forouzan, Data Structures: A Pseudocode Approach With C, Thomson Brooks/Cole Publications, Course Technology, ISBN: 9780534390808.
4. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, Data Structure using C, Prentice- Hall, ISBN: 9780131997462.
5. Robert Kruse, Tondo C L and Bruce Leung, Data Structures & Program Design in C, Pearson India, 2nd Edition, ISBN: 9788177584233.
6. U. A. Deshpande and O. G. Kakde, Data Structures & Algorithms, ISTE Learning Materials Centre, New Delhi, ISBN: 9788188057054.
7. 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.
8. Seymour Lipschutz, Data Structures With C, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070701989.
9. Jean-Paul Tremblay, Paul G. Sorenson, P. G. Sorenson, Introduction to Data Structures with Applications, 2nd Edition, Mcgraw-Hill College, ISBN: 0070651574.

CSS1C03 – THEORY OF COMPUTATION

Objective

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

Course Outcome

- Understand the preliminaries of theory of computation
- Understand the concept of Finite Automata and its categories
- Construct Regular Expressions and identify the concept of DFA minimization

- Identify different types of grammars
- Understand computability, decidability and classify P and NP problems

Course Outline

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 Context Sensitive Grammar (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:

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

CSS1C040 - The Art of Programming Methodology

Objectives:

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

Course Outcome

- Learn the concept of programming
- Understand the concepts of data structures and functions in Python
- Understand the packages in Python for Data Science & Statistical Data Analysis
- Make the students equipped to solve mathematical or scientific problems using Python

Course Outline

Unit I: Part A: Problem Solving - Flow Chart for Structured Programming - Program Charts System Charts - Variables, data names, programming statements - Flow Chart Symbols - Terminal Symbols - I/O - Comments - Connectors - Process - Decision – Loops, Flow Charts of Fundamental Algorithms (mentioned in Part B). Part B: Algorithm Design, Problem Solving Aspect - Top down Design - Formal Conventions Writing Algorithms - Fundamental Algorithms (Discuss the Design of Algorithms only). Part C: Program, Characteristics of a good program - Modular Approach - Programming style - Documentation and Program Maintenance - Compilers and Interpreters - Running and Debugging Programs - Syntax Errors - Run-Time Errors - Logical Errors - Concept of Structured Programming.

Unit II: Features of Python, The Basics: Literal, Constants, Numbers, Strings, Identifier, Naming, Data Types, Objects, Output. Operators and Expressions: Operators, Operator Precedence, Order of Evaluation, Associativity, Using Expressions, Control Flow: Decision making, if, if …else, if…elif…else. Loops: while, for, break, continue statements.

Unit III: Functions: Definition, parameters, local and global variables, default arguments, keyword arguments, return statement, Recursive functions, Lamda functions. UNITs: built in UNITs, creating UNITs, import statement, Namespaces and scope, dir (), reload () functions. File Handling: Opening a file, writing to a file, closing a file, file renaming, deleting a file. Directories of Python: methods mkdir (), chdir (), getcwd (), rmdir ().

Unit IV: Data structures: Introduction, List, storing many different data points under a single name, creates subset and manipulates Lists in all sorts of ways. Tuple, Dictionary, Sequences. Quick introduction to Objects and Classes: Creating objects, init () method, Oops principles: encapsulation, data hiding, Inheritance, method overriding, polymorphism. Exception Handling: try….except, try…finally, raising an exception, user defined exception.

Unit V: Python in Data Science: Introduction, Python packages for data science: Pandas, Numpy, Scipy, Matplotlib, Python Data Operations: Importing and Exporting data, Data Cleansing, Processing

CSV Data, Processing JSON Data, Processing XLS Data. Python Data Visualization: Charts, Chart Styles, Scatter plots, 3D Charts, Time Series, Geographical Data. Statistical Data Analysis: Measuring Central Tendency, Measuring Variance, Correlation in Python.

References:

1. Introduction to Programming Using Python, First Edition by Y. Daniel Liang, Prentice Hall, ISBN-100132747189100
2. Learning Python, David Ascher and Mark Lutz,O'Reilly Publications.
3. Python for Dummies,Aahz Maruch and Stef Maruch
4. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython Samir Madhavan, Mastering Python for Data Science

CSS1C05 Computer Organization Architecture

Objectives: To understand the basics of Computer System Architecture

Course Outcome

- Understand the number systems, Boolean algebra and flip flops
- Understand the basic computer organization
- Understand the working of ALU
- Identify different types of memory
- Understand the architecture of 8085 microprocessor

Course Outline

Unit I [16 T]: 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[12 T]: 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, micro programmed control, micro instructions-types.

Unit III [14 T]: 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.

Unit IV [12 T]: Main Memory - memory hierarchy - main memory - RAM, ROM - memory cells cell organization - working - performance considerations - cache memory - virtual memory

memory management requirements - secondary storage - memory interleaving. Input / Output Organization - Accessing I/O ,d&Vices - programmed I/O, interrupt I/O - interrupts - interrupt processing - hardware interrupts - programmable interrupt controller - vectored interrupts - interrupt nesting - daisy chaining - direct memory access (DMA) - DMA operations & DMA Controller, Introduction to I/O interfaces, I/O channels, IO Processors.

Unit V [18T]: 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 estimation of execution time - estimation of execution time. Intel 8051 Micro controller - Architecture - basic instructions - basic assembly language programs peripherals: interrupts, timers, parallel port, serial port.

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 and *et al.*, *The 8051 Microcontroller and Embedded Systems*, Pearson Education Asia, 5th Indian Reprint, ISBN: 013119402X.

CSS1L01 – PRACTICAL I

Objectives

- To practically implement the theory portions covered in Art of Programming Methodolgy (CSS1C04) and Advanced Data Structures (CSS1C02).

Course Outline

Unit I: Python Programming

1. Demonstrate data types in Python
2. Demonstrate if-else in Python
3. Demonstrate Switch in Python
4. Demonstrate loops in Python
5. Demonstrate Set Operations in Python
6. Demonstrate Arrays in Python
7. Demonstrate Lists in Python
8. Demonstrate Dictionaries in Python
9. Demonstrate Strings in Python
10. Demonstrate OOPs in Python
11. Demonstrate files in Python
12. Demonstrate importing and exporting data in Python
13. Demonstrate missing data analysis in Python
14. Demonstrate drop columns in Python
15. Demonstrate reading CSV file in Python
16. Demonstrate reading XLS file in Python
17. Demonstrate creating charts in Python
18. Demonstrate calculation of mean, median and mode in Python
19. Demonstrate calculation of standard deviation in Python
20. Demonstrate correlation using Python

Unit 2: Data Structures and Algorithms

1. Implementation of stacks using arrays.
2. Implementation of queues, circular queue using arrays.
3. Implementation of sequential search and binary searchtechniques.
4. Implementation of linked lists and operations (add, insert, delete, search) on linked lists.
5. Implementation of stacks using linked list.
6. Implementation of queues using linked list.
7. Implementation of doubly linked list.
8. Implementation of circular linked list.
9. Implementation of binary tree and traversals.
10. Implementation of Binary search trees and perform the operations on BST.
11. Implementation of various sorting algorithms.
12. Conversion of an infix expression to the postfix form using stacks.

13. Evaluation of a postfix expression.
14. Implementation of graphs and graph traversals.
15. Implementation of heap tree and operations.

CSS1A01 – INTRODUCTION TO RESEARCH (ABILITY ENHANCEMENT AUDIT COURSE)

Objectives

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

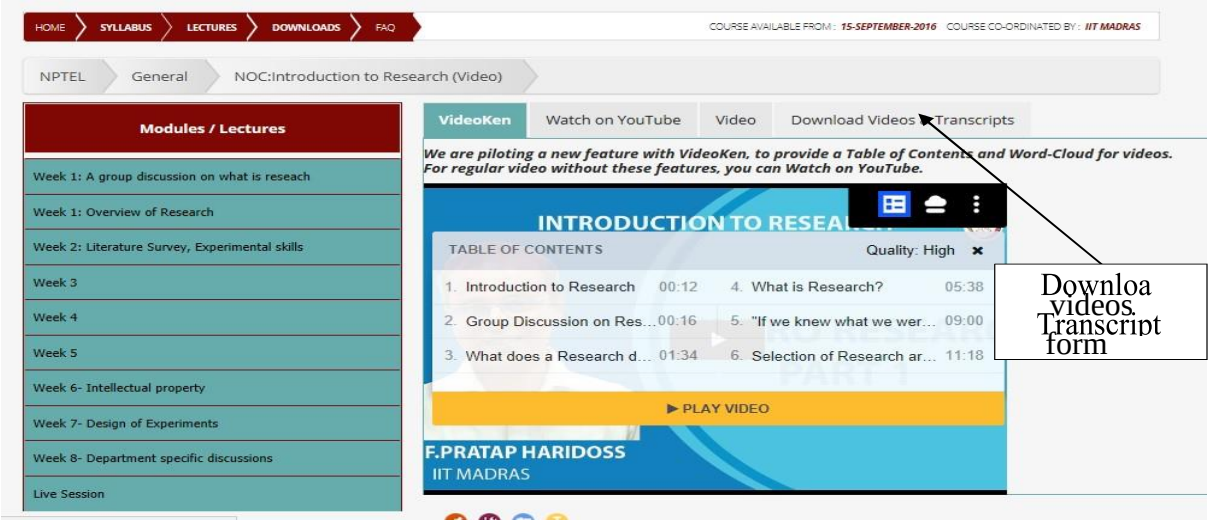
- Understand research terminology
- Be aware of the ethical principles of research
- Identify the components of a literature review process
- Critically analyse published research
- To introduce research methods in the field of computer Science

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 examination of 2 Hrs duration with a minimum of 20 weightage before the conclusion of first semester classes and have to intimate /upload the results of the same to the University on the stipulated date during the III Semester. Students have to obtain only minimum pass requirements in this Audit Course.

Course Delivery Mode

This course is an Ability Enhancement Audit Course. The course content is not delivered in the classrooms. Instead, the students have enroll 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 Outline

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

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

References:

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

SEMESTER II

CSS2C06 – DESIGN AND ANALYSIS OF ALGORITHMS

Objectives:

- 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.

Course Outcome

- Understand the basics of design of algorithms and various models of computation
- Understand the analysis of different approaches in algorithm design
- Understand Asymptotic notations and analysis of solving recurrences
- Identify different complexity classes
- Analyze parallel algorithms and their efficiency

Course Outline

Unit I: 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.

Unit II: 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).

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

Unit IV: 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.

Unit V: 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, Deterministi c 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 (Unit I, II, III and IV).
4. Sanjay Razdan, Fundamentals of Parallel Computing, Narosa Publishing House, 2014, ISBN: 9788184873481 (Unit V).
5. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013, ISBN: 9788131803349 (Unit I, II, III and IV).
6. Upadhyay N, Design and Analysis of Algorithms, SK Kataria & Sons, 2008 (Unit I, II, III and IV).
7. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley, ISBN: 9780201003277 (Unit I, II, III and IV).
8. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India, ISBN: 0133350681 (Unit I, II, III and IV).
9. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill, ISBN: 0070237530 (Unit I, II, III and IV).
10. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd, ISBN: 8175152575 (Unit I, II, III and IV).
11. Oded Goldreich, P, NP and NP - Completeness, Cambridge University Press, 2011. ISBN: 0521122546 (Unit V).
12. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms, Volume 1, Addison Wesley, 1997, ISBN: 8177587544 (Unit I).
13. Sanjeev Arora and Boaz Borak, Computational Complexity - A Modern Approach, Cambridge University Press; 2009, ISBN: 0521424267 (Unit III).
14. Daniel Hills W and Bruce M Boghosian, Parallel Scientific Computation, Science, 13 August 1993, Vol. 261 (5123), pp.856-863 (Unit V).

CSS2C07 – OPERATING SYSTEM CONCEPTS COURSE

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

Course Outcome

- Understand the basics of Operating Systems
- Understand the concept of concurrency
- Understand the concept of memory management
- Understand different scheduling algorithms
- Analyze the features of variety of operating systems

Course Outline:

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

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

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

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

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

Reference

1. William Stallings, *Operating System- Internals and Design Principles*, 7th Edition, Pearson, ISBN: 9780273751502.
2. Abraham Silberschatz, Peter B. Galvin and, Greg Gagne, *Operating System Concepts*, 9th Edition, John Wiley & SonsTISBN: 9781118063330.
3. Ann Mclver McHoes and Ida M. Flynn, *Understanding Operating Systems*, 6th

Edition, Cengage Learning, 2010, ISBN: 9781439079201
4. Mukesh Singhal and Niranjan G. Shivaratri, *Advanced Concepts in Operating Systems - Distributed, Database, and Multiprocessor Operating Systems*, Tata McGraw-Hill Education Private Limited, ISBN: 9780070575721.
5. Current Literature (for Mobile Operating Systems).

CSS2C08 – COMPUTER NETWORKS

Objectives:

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

Course Outcome

- Summarize the concepts of fundamentals of computer networks
- Identify categories of communication media and analysis of ISO - OSI model and TCP/IP model
- Understand the concept of Application layer protocols
- Identify transport and network layer protocols and analyze various routing algorithms
- Understand Link layer services and multiple access protocols
- Understand the basic concepts of security in computer networks

Course Outline:

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

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

Unit III: 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 – mobility.

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:

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

9789812359865 2000.
5. Andrew S. Tanenbaum, *Computer Networks*, 5th Edition, PHI, ISBN: 9788131787571.
6. Herbert Scheldt, *Java Complete Reference*, 7th Edition, McGraw-Hill Osborne Media, ISBN: 9780072263855.

CSS2C09 - Computational Intelligence

Objective

To understand the basics of Artificial Intelligence and its applications

Course Outcome

- Concepts of Artificial intelligence and its applications
- Apply different searching techniques for problem solving in Artificial intelligence
- Learning the concepts of knowledge representation and reasoning using Artificial intelligence
- Understand and apply the concepts of game playing using Artificial Intelligence.
- Understanding the concepts of expert system and to create expert systems to solve complex problems of a particular knowledge domain.
- Understanding various concepts of machine learning
- Apply machine learning to solve problems in real life examples

Course Outline

Unit I: [10 T]

Introduction - Artificial Intelligence - problems, scope and applications, problem space and search - production system- characteristics – artificial intelligent agents, structures and strategies for state space search, strategies for space search, using state space to represent reasoning with the predicate calculus.

Unit II: [15 T]

Search technique: depth first search, breadth first search. 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: [18 T]

Knowledge representation issues, representation and mappings, representing simple facts in logic, representing instances and ISA relationships, computable functions and predicates, first order predicate calculus, inference rules, clause form, resolution, unification. natural deduction, knowledge representation using rules, Slot and filler structures: semantic nets, frames, conceptual dependency, scripts, logic programming, forward versus backward reasoning, symbolic reasoning under uncertainty- non-monotonic reasoning.

Unit IV: [18 T]

Game playing - the mini-max search procedure, adding alpha-beta cut-offs, additional refinement, iterative deepening, planning system and its components, understanding, understanding as constrained satisfaction. Definition and characteristics of expert system, representing and using domain knowledge, expert system shells. Knowledge engineering, knowledge acquisition, expert system life cycle & expert system tools, MYCIN & DENDRAL examples of expert system.

Unit V: [11 T]

Machine learning - rote learning, learning by taking advice, learning in problem solving, learning from examples, explanation based learning, analogy, formal learning theory, connectionist models -

hopfield networks, learning in neural networks, back propagation, the genetic algorithm, classifier systems and genetic programming, artificial life and society based learning.

References:

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

CSS2C10 – PRINCIPLES OF SOFTWARE ENGINEERING

Objectives:

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

Course Outcome

- Understand the fundamental concepts of Software engineering, process models
- Understand the Agile model
- Understand the process of requirement engineering
- Identify various testing strategies and maintenance activities

Course Outline

Unit I: Introduction – problem domain - software engineering challenges – approaches – software process and development models – agile models – SDLC - software process.

Unit II: Software requirements analysis & specification - feasibility study - types of feasibility – software requirements - problem analysis – requirement specification – functional specification – metrics. Software design – outcome – cohesion and coupling – layered arrangement of modules – approaches to software design - structured analysis – DFD -extending DFD technique for applying to real-time systems – structured design – detailed design - object oriented modelling – use case model – class diagram – interaction diagram - activity diagram - data diagram – state chart diagram - ER diagram.

Unit III: User Interface (UI) design – characteristics – basic concepts – types – fundamentals of component-based GUI Development – UI design methodology – process planning – cost estimation – project scheduling – configuration management – risk management - software coding – review – documentation – software testing - software testing basics - steps involved in test plan - software testing strategies.

Unit IV: Managing project – time management – setting aims and objectives – techniques for generating ideas – literature survey – types of information sources – writing literature survey.

Unit V: Project story preparation – key deliverables – communicating with experts – forms of

communication – presenting ideas – common problems faced by a research scholar – report writing.

References:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, ISBN: 9788173197024.
2. Rajib Mall, Fundamentals of Software Engineering, 3rd Edition, PHI Learning Pvt Ltd, ISBN: 9788120338197.
3. Rohit Khurana, Software Engineering: Principles and Practices, 2nd Edition, Vikas Publishing House Pvt Ltd, ISBN: 8125939466.
4. Andy Hunt, Your Research Hunt, How to Manage it, Routledge, ISBN: 0415344085.
5. Michael Jay Polonsky, David S. Waller, Designing and Managing a Research Project: A Business Student's Guide, Sage, ISBN: 1412977754.
6. Richard Bullock, Maureen Daly Goggin and Francine Weinberg, The Norton Field Guide to Writing (with Readings and Handbook), 3rd Edition, W. W. Norton & Company, ISBN: 0393919595.
7. Kavadia Garg, Agrawal and Agrawal, An introduction to Research Methodology, Rbsa Publishers ISBN: 8176111651.

CSS2L02 – PRACTICAL II

Objectives: To practically implement concepts in Networks and Operating Systems.

Course Outline

Unit I: 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. Establish a LAN that consists of at least one server and two clients*.
3. Study of network utilities in Linux/Windows (hostname, ping, ifconfig, ipconfig, netstat, nslookup, telnet, traceroute, finger, telnet, tracert, arp, ftp etc)*.
4. Implementation of TCP Client.
5. Implementation of TCP Server.
6. Write a program to check the Date and Time in TCP Date Time Client.
7. Write a program to check the Date and Time in TCP Date Time Server.
8. Implementation of UDP client and server.
9. Write a program to transfer Files using UDP.
10. Implementation of transferring files using FTP.
11. Write a program to simulate the sliding window protocol.
12. Study of Network Simulators (NS2/Glomosim)*.

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

Unit II: Operating System Concepts

1. Write programs using the following system calls: fork(), execl() and wait().
2. Write File System Calls to write, append and display.
3. To accept the burst time for a set of processes for FCFS scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.
4. To accept the burst time for a set of processes for SJF scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.

5. To accept the burst time and priority for a set of processes for Priority scheduling and
6. create chart consisting of the burst time, priority, turnaround time and wait time of each process.
7. To create n Fibonacci numbers and prepare a list of prime numbers amongst them (use pipe for IPC).
8. To demonstrate IPC using shared memory.
9. 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.
10. 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.
11. To demonstrate the process of contiguous allocation of memory blocks to store files of varying sizes.
12. To implement Producer Consumer problem using semaphores.

CSS2A02 – TERM PAPER (PROFESSIONAL COMPETENCY AUDIT COURSE)

Objectives:

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

The student is expected to do an extensive literature survey and analysis in an area related to Computer Science, chosen by him/her, under the supervision of a faculty member from the department. The student has to choose an area for his/her work after due consultation and approval from the guide. The study should preferably result in a critical review of the present works/design ideas/designs/algorithms/theoretical contributions in the form of theorems and proofs/new methods of proof/new techniques or heuristics with analytical studies/implementations and analysis of results.

The student should give a seminar on his/her work, during the semester, and submit a technical report. Technical report should be prepared in IEEE conference style format.

Course Delivery Mode

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

Course Evaluation & Course Credit

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

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

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

References:

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

SEMESTER III

CSS3C11 – ADVANCED DATABASE MANAGEMENT SYSTEM

Objectives:

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

Course Outcome

- Understand the concept of database
- Understand the design of database
- Understand the concept of SQL
- Understand concurrency control
- Compare the features of different databases

Course Outline

Unit I: Introduction - purpose of database systems, views of data - data abstraction, instances and schemas, data independence, data models - hierarchical data model, network data model, relational data model, ER diagram. Database languages - DDL, DML, transaction management, storage management, database administrator, database users, overall system structure. Relational data model - relational model concepts, keys, integrity constraints - domain constraints, key constraints, entity integrity constraints, referential integrity constraints. ER data model - basic concepts, constraints, keys, design issues, entity relationship diagram, weak entity sets, extended ER features, design of an ER database schema, reduction of an ER schema to tables. Relational algebra and calculus - relational algebra - selection and projection, set operations, renaming, joins, division. Relational calculus - tuple relational calculus, domain relational calculus. Expressive power of algebra and calculus.

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

Unit III: Relational database query languages - basics of QBE and SQL. Data definition in SQL data types, creation, insertion, viewing, updation, deletion of tables, mo difying the structure of the tables, renaming, dropping of tables. Data constraints - I/O constraints, primary key, foreign key, unique key constraints, ALTER TABLE command database manipulation in SQL - computations done on table data - SELECT command, logical operators, range searching, pattern matching, grouping data from tables in SQL, GROUP BY, HAVING clauses. Joins - joining multiple tables, joining a table to it. DELETE - UPDATE. Views - creation, renaming the column of a view, destroys view. Program with SQL - data types Using SET and SELECT commands, procedural flow, IF, IF /ELSE, WHILE, GOTO, global variables. 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 (Use MySQL as the RDBMS).

Unit IV: 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, deadlock, granularity, timestamp ordering protocol. Basics of query processing.

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

References

1. Elmasri and Navathe, *Fundamentals of Database Systems*, 5th Edition, Pearson, ISBN: 9788131758984.
2. Abraham Silbersehatz, Henry F. Korth and S.Sudarshan, *Database System Concepts*, 6th Edition, Tata McGraw-Hill, ISBN: 0071325220.
3. CJ Date, *An Introduction to Database Systems*, 8th Edition, Addison Wesley, ISBN: 0321197844.
4. Ramakrishnan and Gehrke, *Database Management Systems*, 3rd Edition, McGraw - Hill Education, ISBN: 9339213114.
5. Alexis Leon and Mathews Leon, *Database Management Systems*, 1st Edition, Vikas Publishers, ISBN: 8182092221.
6. Vikram Vaswani, *MySQL The complete Reference*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070586845.
7. Joel Murach, *Murach's Mysql*, Mike Murach & Associates Inc, ISBN: 9350237695.
8. Paul DuBois, *MySQL Cookbook*, 2nd Edition, O'Reilly Media, ISBN: 8184042809.

CSS3C12 – OBJECT ORIENTED PROGRAMMING CONCEPTS

Objectives: To learn object oriented concepts and programming concepts and methodologies and to learn its implementation using Java.

Course Outcome

- Understand the fundamental concepts of OOPS
- Apply the basic programming constructs in Java
- Understand various tools to make interactive GUI using Java and implement event driven programming
- Apply programming in JDBC

Course Outline

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 - FOR, IF, WHILE, DO WHILE, SWITCH, BREAK, CONTINUE statements - operators - casts and conversions - arrays.

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

Unit III: 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 (Use MySQL as the RDBMS). Sockets: introduction to networking, InetAddress, url, socket, server sockets, datagrams.

Introduction to Unified Modelling Language (UML), 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

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. Kathy Sierra, *Head First Java*, 2nd Edition, Shroff Publishers and Distributors Pvt Ltd, ISBN: 8173666024.
4. David Flanagan, Jim Farley, William Crawford and Kris Magnusson, *Java Enterprise*
5. *in a Nutshell: A Desktop Quick Reference*, 3rd Edition, O'Reilly Media, ISBN: 0596101422.
6. Grady Booch, James Rumbaugh and Ivar Jacobson, *The Unified Modeling Language User Guide*, 2nd Edition, Pearson, ISBN: 8131715825.

CSS3C13 – Principles of Compilers

Objective

To understand the basics of Compilers and the different steps involved in generating machine code.

Course Outline

- Understand the basics of various system software
- Analyze various phases of compilation
- Understand the lexical phase of compilers
- Understand the concept of parsing and its different types
- Identify immediate code generation issues and various representation forms
- Understand the concept of Runtime environments and storage optimization strategies
- Identify Code generation issues and analyze various techniques for code optimization

Course Outline

Unit I [12 T]: Introduction to compiling - definition of compiler, translator, interpreter, analysis of the source program, the phases of a compiler, compiler construction tools- applications of compiler technology – programming language basics - lexical analysis – role of lexical analyser input buffering - specification of tokens – recognition of tokens using finite automata - regular expressions and finite automata - from NFA to DFA - Regular Expression to an NFA Design of a lexical analyser generator.

Unit II[12 T]: 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 – ambiguity – associativity and precedence of operators - writing a grammar – top- down parsing – recursive descent parsing - FIRST and FOLLOW – LL (1) Grammars – recursive predictive parsing - bottom up parsing – reductions – handle pruning – shift reduce parsing - operator precedence parsing, simple LR parsing.

Unit III [12 T]: Intermediate code generation – DAG – three address code – addresses and instructions – quadruples – triples – Static Simple Assignment form – types and declarations – typeexpressions-typeequivalences–declarations–typechecking–rules–typeconversion – function and operator overloading – type inference and polymorphic functions – control flow – boolean expressions – short circuit code – flow-control statements – control-flow translation for boolean expressions – BREAK CONTINUE and GOTO statements.

Unit IV [12 T]: Run time environments – storage optimization – static Vs dynamic allocation – stack allocation of space - activation trees and records – calling sequences – access to non local data on the stack – data access without nested procedures – issues with nested procedures – heap management – the memory manager – the memory hierarchy – locality in programs – reducing fragmentation - manual deallocation requests.

Unit V [12 T]: 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. 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).

CSS3L03 – PRACTICAL III

Objectives: To practically implement DBMS constructs and OOPS

Course Outline

Unit I: Advanced Database Management System

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

Unit II: Object Oriented Programming Concepts

1. Simple Java programs like computing formulas expressions.
2. Programs involving loops and decisions like generating Fibonacci, prime, strange series.

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

CSS3E01a – DATA MINING & FEATURE ENGINEERING

Objectives:

- To understand the basics of data and data mining
- To understand the different tools used in visualization

Course Outcome

- Understand the science of data and basic data mining concepts
- Understand the need of visualization and its application strategies
- Understand the need of feature engineering

Course Outline

Unit I: Introduction to Data Science: Data Science, Data Scientist Vs Data Analyst, Data Analytics Life Cycle: Discovery, Data Preparation, Model Planning, Model Execution, Communicate Results, Operationalize.

Introduction to Data mining: Meaning, Definition, Goals, Scope, Related technologies, Stages involved in data mining, Data mining techniques , Major issues in data mining, Applications.

Unit II: Introduction to data: Data objects and attribute types, Measures of central tendency and dispersion, attribute generalization and relevance, Class comparison, Statistical measures, Measures of similarity and dissimilarity; Data pre-processing: Overview, Data cleaning, Data integration, Data reduction, Data transformation, Discretization, Generating concept hierarchies.

Unit III: Feature Engineering: Vectors, Orthogonal vs orthonormal vectors, Eigen values and vectors, Subset Selection, PCA, Feature Embedding, Factor Analysis, Singular Value Decomposition and Matrix Factorization, Linear Discriminant Analysis, Canonical Correlation Analysis.

Unit IV: Mining frequent patterns, associations and correlations: Basic concepts and methods, Frequent itemset mining methods - Apriori algorithm, Pattern growth approach, Framing association rules, Pattern evaluation methods, Pattern mining concepts, Mining in multi-level and multidimensional space, Constraint based frequent pattern mining.

Unit V Data Visualization Tools and its applications: Multidimensional Visualization Tools: Box plot, histogram, Quantile plot, QQ Plot, Heat map, Scatter Plots, Contours, Glyphs, Parallel Plot, tree maps. Implementation of Visualization tools using Python/ Orange

References:

1. Jiawei Han, Micheline Kamber and Jian Pei, *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, 2011
2. Pang-NingTan, Michael Steinbach and Vipin Kumar, *Introduction to Data Mining*, Pearson Education Inc , 2003
3. Ethem Alpaydin, *Introduction to machine learning*, MIT Press,2014
4. Alex Berson and Stephen J. Smith, *Data Warehousing*, *Data Mining & OLAP*, *Computing*Mcgraw-Hill, Tata McGraw-Hill Education, 2004
5. K.P. Soman, ShyamDiwakar and V. Ajay, *Insight into Data mining Theory and Practice*, Prentice Hall of India, 1st Edition
6. G. K. Gupta, *Introduction to Data Mining with Case Studies*, PHI Learning Pvt. Ltd, 3rd Edition.

CSS3E01b – INTRODUCTION TO DATA WAREHOUSE

Objectives

- Acquaint with concepts, architectures, design of data warehouse
- Understand the types of data warehouses and their applications

Course Outcome

- Understand the characteristics of data warehouses and various data schemas
- Understand the architecture and implementation of data warehouses
- Acquaint with advanced techniques in data warehousing

Course Outline

UNIT I: Introduction to data warehouse, Types of data: Subject-oriented, Integrated, Time-variant, Nonvolatile data, Data granularity, Data Marts, Data Modelling: Star Schema: fact table, dimension table, Snowflake schema, fact constellation, OLAP operations on multi-dimensional data model.

UNIT II: Data Warehouse Architecture: Data ware house view: top-down, data source, data warehouse, business query view. Process of Data Warehouse design, 3 tier data warehouse architecture, data warehouse back-end tools and utilities, Meta data repository. Types of OLAP servers: ROLAP, MOLAP, HOLAP Data Warehouse Implementation: Computation of Data cubes, partial materialization, Indexing OLAP data, Processing of OLAP queries.

UNIT III: Conceptual Modelling: Conceptual Modelling of Data warehouse, Logical modelling of Data warehouses, Physical Data warehouse design, Extraction, Transformation, and Loading: Business Process Modeling Notation, Conceptual ETL using BPMN.

UNIT IV: Data ware house Analytics: Data Mining Tasks in data warehouse: Classification, Clustering, Association Rules, Pattern Growth algorithm, Sequential Patterns.

UNIT V: Advanced Techniques: Spatial Data warehouse: Spatial data types, conceptual modelling of spatial data ware house, Implementation. Trajectory data warehouse: Mobility Data Analysis, Temporal Data Types Implementation of Temporal data warehouse, Big data warehouse Technologies: map-Reduce and Hadoop, Column-Store Database system, In-Memory Database System, ETL in big data.

References:

1. Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann Publishers
2. Data Warehouse Systems: Design and Implementation, Alejandro Vaisman, Esteban Zimanyi, 2014, Springer
3. DATA WAREHOUSING: Concepts, Techniques, Products and Applications, C.S.R. PRABHU, 3rd Edition, PHI Learning
4. Data Architecture: A Primer for the Data Scientist: Big Data, Data Warehouse, W.H. Inmon, Daniel Linstedt, 2015, Morgan Kauffman

CSS3E01c – FUNDAMENTALS OF BIG DATA

Objective

To understand the concept of different operational databases used for big data

Course Outcome

- To understand the characteristics and concepts of big data
- To familiarize with cutting edge technologies of big data technology

Course Outline

Unit I: Introduction to Big Data : Definition & importance of Big Data ,Four dimensions of big data, Volume, Velocity, Variety, Veracity, Importance of big data, Structured data, Unstructured data, The role of a CMS in big data management, Integrating data types into a big data environment, Distributed computing and big data; Big data stack: layer 0,1 and 2; Big data management, Operational databases ,Relational databases ,Non-relational databases, NoSQL, Key-value pair databases ,Document databases, Columnar databases, Graph databases ,Spatial databases.

Unit II: Big Data analysis: Basic analytics, Operationalized analytics, Modifying business intelligence products to handle big data, Big data analytics examples, Analytics solutions, Text analytics, Exploring unstructured data, Understanding text analytics, Analysis and extraction techniques, The extracted information, Text analytics tools for Big Data, Custom applications for big data analysis, R

environment, Google prediction API, Characteristics of a big data analysis framework.

Unit III: NoSQL databases: Types, Advantages over relational databases; MongoDB: Introduction, MongoDB philosophy, The data model, Designing the database, Collections, Documents, Data types,The _id Field, Indexes, Viewing available databases and collections, Opening a database, Inserting data, Querying for data, Retrieving documents, Aggregation commands, Grouping results, Conditional operators, Specifying an array of matches, Applying criteria for search , $slice , $size , $exists , $type , $elemMatch , $not (meta-operator), update() , save() , $inc, $set, $unset, $push, $pushAll,$addToSet, Removing elements from an array, Atomic operations, Modifying and returning a document atomically, Renaming a collection, Removing data, Referencing a database, Implementing index, Related functions , min() and max().

Unit IV: Hadoop : History, Components, HDFS, MapReduce basics, Origins of MapReduce, Map function, Reduce function, Putting them together; Hadoop common components, Application development in Hadoop, Pig and Pig Latin, Load, Transform, Dump and store, Hive, Jaql, Getting our data into Hadoop, Basic copy data, Flume, Zookeeper, HBase ,Oozie, Lucene, Avro.

Unit V: Apache Spark: Introduction, Spark Architecture, RDD, Components of Spark, Advantages, Core and significance, Data Frame and Dataset, Core Programming, Spark SQL, Submitting Jobs in Yarn, Deployment, Advanced Spark Programming,, Machine Learning with MLlib, Spark Streaming.

References:

1. Hurwitz, Alan Nugent, Fern Halper and Marcia Kaufman, *Big Data for Dummies* , John Wiley & Sons, 2013
2. EelcoPlugge, Peter Membrey and Tim Hawkins ,*The Definitive Guide to MongoDB: TheNOSQL Database for Cloud and Desktop Computing*, Apress, I Edition
3. Chris Elaton, Derk Deroos, Tom Deutsch, George Lapis and Pual Zikopoulos, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, McGraw Hill Professional, 2011
4. Garry Turkington, *Hadoop Beginner's Guide*, Packt Publishing Ltd, 2013
5. Holden Karau, Andy Konwinski, Patrick Wendell & Matei Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*, 1st Edition

CSS3E01d – STATISTICAL TECHNIQUES FOR DATA SCIENCE

Objective

To understand the basics of Statistics used in Data Science

Course Outcome

- Understand the basic concepts of Statistics used in Data Science
- Know-how of types of data and hypotheses used in data analytics

Course Outline

Unit I: Probability Distributions: Definition classical, empirical, Addition and multiplication laws, Conditional probability, random variable, Probability functions, discrete distribution-Binomial distribution, Poisson distribution. Continuous distribution- Normal distribution, lognormal distribution. Exponential Distribution (Concept and applications only).

Unit II: Sampling Distributions: Theory of sampling distributions, Standard error, Sampling distribution of Sample mean, Chi square distribution, Student's t distribution, F distribution. (Concept and applications only)

Unit III: Statistical Estimation: Point estimation, Properties of point estimation, Unbiasedness, consistency, sufficiency, efficiency. Interval estimation, Confidence intervals.

Unit IV: Testing of Hypothesis: Test of hypothesis- Null and alternative hypothesis, Type I and Type II errors, Critical region, Level of significance, *P Value*, Power of a test, Critical value, Neymann Pearson Lemma.

Unit V: Parametric and Non parametric Tests: Parametric Tests- t, F, Z tests. Non Parametric Tests – Kolmogrov Smirnov test, Kruskal Wallis test, Mann Whitney U test, Wilcoxon rank sum test, Welch's t test, Plots to check normality. Analysis of variance and Covariance.

References:

1. S.C.Gupta and V.K.Kapoor (2010) *Fundamentals of Mathematical Statistics*, Eleventh thoroughly revised edition, Sultan Chand & Sons, New Delhi
2. Sheldon Ross (2014), *A first course in Probability*, Ninth edition, Pearson Education Inc.
3. E.L. Lehmann and Joseph P. Romano (2005), *Testing Statistical Hypothesis*, Springer.
4. George Casella and Roger L. Berger (2001), *Statistical Inference*, 2nd Edition

CSS3E01e – Computing Approaches

Objective

To make students analyze about different computing approaches through distributed, cloud and parallel computing techniques.

Course Outcome

- Understand the various computing paradigms used in Computer Science
- A basic knowledge on Socket programming

Course Outline

Unit I: Overview of Computing Paradigm: Definition, types, Recent trends in Computing; Introduction to distributed programming: Anatomy of a Distributed Application, Requirements for Developing Distributed Applications.

Unit II: Introduction to sockets programming : Sockets and Streams, Distributing Objects, CORBA, Java RMI,RMI vs. CORBA, Thread, Message passing systems and its need, Peer-to-peer communication and group communication, Service Oriented Architecture.

Unit III: Distributed systems: Characteristics, challenges, clocks, events and process status, synchronizing physical clocks, logical time and logical clocks, global states, Distributed Mutual Exclusion, Distributed concurrency control, Distributed deadlock.

Unit IV: Cloud Computing: Evolution of cloud computing, Comparison with traditional computing architecture (client/server), Services provided at various levels, Role of Networks in Cloud computing, Role of Web services; Service Models (SaaS): Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS); Deployment Models: Public cloud, Private cloud, Hybrid cloud, Community cloud, Cloud Security.

Unit V: Parallel Computing: Flynn's Classification of Computer Architecture, Types of Parallelism, Parallel programming models.

References:

1. Robert Orfali & Dan Harkey, Client/Server Programming With Java And Corba, 2nd Ed
2. Jim Farley, O'Reilly, *Java Distributed Computing.*
3. Malhar Barai , Vincenzo Caselli, Binildas A. Christudas, , *Service Oriented Architecture With Java*, Adobe Press, 2005
4. Qusay H. Mahmoud, *Distributed Programming with Java*, Manning Publisher 2000.
5. Barrie Sosinsky, *Cloud Computing Bible*, Wiley-India, 2010
6. George Couluris, Jean Dollimore, *Distributed Systems Concepts & Design*, Pearson education, 3rd edition, 2006.
7. Peter Pacheco, An Introduction to Parallel Programming

CSS3E02a – MACHINE LEARNING

Objectives:

- To develop an appreciation for what is involved in learning from data.
- To understand a wide variety of learning algorithms.
- To understand how to apply a variety of learning algorithms to data
- To understand how to perform evaluation of learning algorithms and model selection.

Course Outcome

- Acquaint with the basics of Machine learning algorithms
- Applied knowledge to build a machine learning model

Course Outline

Unit I: Techniques of Machine Learning: Supervised learning, Unsupervised learning, Semi supervised learning, Reinforcement learning, Active learning, Transfer learning

Unit II: Classification: Basic concepts, Decision tree induction, Bayes classification methods, Rule based classification, K-nearest neighbors, Ensemble methods, Bagging, Boosting & Random Forests. Evaluation measures for classification.

Unit III: Clustering: Partitioning methods: k-means, k-medoids; hierarchical clustering: Chameleon, BIRCH; density based methods: DBSCAN, OPTICS; Grid Based method: STING, CLIQUE. Expectation-Maximization Algorithm. Clustering evaluation measures: Extrinsic and intrinsic methods.

Unit IV: Regression in Machine Learning: Linear regression- univariate, multi variate models, Logistic Regression, Discriminant Analysis, Poisson Regression, Regularization methods- ridge, lasso, and gradient descent.

Unit V: Non Parametric techniques: SVM, ANN, Bayesian Belief Networks, Gaussian Process. Case studies of classification, clustering using parametric/non-parametric approaches in web mining and text mining

References:

1. Bishop C, *Pattern Recognition and Machine Learning*, Springer 2007
2. Rogers S and Girolami M, *A first course in Machine Learning*, CRC Press, 2011
3. Cory Lesmeister, *Mastering Machine Learning with R*, Packt, 2nd Edition
4. Mitchell T, *Machine Learning*, McGraw-Hill, 1997
5. Barber D, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012
6. Duda, Hart and Stork, *Pattern Classification*, Wiley-interscience,2012
7. Sinan Ozdemir, Divya Susarla, *Feature Engineering Made Easy: Identify unique features from your dataset in order to build powerful machine learning systems*, Packt Publishing Ltd, 2018

CSS3E02b – PATTERN RECOGNITION

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 pre-processing, feature extraction, and feature selection to multivariate data.
- To understand supervised and unsupervised classification methods to detect and characterize patterns in real-world data.

Course Outcome

- Understand various pattern recognition algorithms used
- Familiarity with various statistical and parametric concepts used in pattern recognition
- Understand the supervised and unsupervised techniques of pattern recognition

Course Outline

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

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

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

Unit IV: Syntactic methods - stochastic search - Boltzmann learning - Nonmetric methods

- decision trees - CART - other tree methods, grammatical methods, grammatical inference.

Unit V: Unsupervised learning and clustering - mixture densities and identifiability, maximum

likelihood estimates, applications to normal mixtures, unsupervised Bayesian learning, data description and clustering.

References:

1. 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:
4. 0138801207.
5. Rajjan Shinghal, Pattern Recognition: Techniques and Applications, 1st Edition, Oxford University Press India, ISBN: 0195676858.

CSS3E02c- Predictive Analytics and Data Modelling

Objective

To understand the basics of Predictive analytics and data modelling

Course Outcome

- Represent and manipulate data in effective ways
- Manipulate data using packages/tools and by ad hoc data handling
- Use mathematical, computational and statistical tools to detect patterns and model performance

Course Outline

Unit I: Correlation Techniques: Data types or levels of measurement- Nominal, ordinal, interval and ratio, Measures of Correlation, Simple correlation, Partial correlation and Multiple correlation.

Unit II: Regression Analysis: Simple linear regression, Gauss Markov theorem, Basics of fitting and residual analysis, Multiple linear regression, model adequacy techniques, Path Analysis (concept only).

Unit III: Logistic regression: Moving from linear to logistic regression, Model assumptions and Odds ratio, ROC curve and KS statistic. (Concept and Applications using real data sets)

Unit IV: Time series analysis: Components of time-series, additive and multiplicative models, Methods for measurement of trends, Methods for measurement of seasonal fluctuations, forecasting, Autocorrelation, ARIMA Model, ARMA Model.(Concept and Applications using real data sets)

Unit V: Multivariate Data Analysis: Multivariate data, plotting multivariate data, matrix scatter plot, Cluster analysis, Principal Component analysis, Discriminant analysis, Factor analysis.
(Concept and Applications using real data sets).

References:

1. D.C.Montgomery, E.A.Peck and G.G.Vining (2006) *Introduction to linear regressionanalysis*, Third Edition, Wiley India Private Ltd, New Delhi.
2. S.C.Gupta and V.K.Kapoor (2014) *Fundamentals of Applied Statistics*, Fourth thoroughly revised edition, Sultan Chand & Sons, New Delhi.
3. P.Mukhopadhyay (2016) *Applied Statistics*, Second Edition (thoroughly revised), Books and Allied (P) Ltd., Kolkata.
4. S.C.Gupta and V.K.Kapoor (2010) *Fundamentals of Mathematical Statistics*, Eleventh thoroughly revised edition, Sultan Chand & Sons, New Delhi.
5. H¨ardle, W. and Simar, L. (2003). *Applied Multivariate Statistical Analysis*

CSS3E02d – VIRTUALISATION AND CLOUD COMPUTING

Objective

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

Course Outcome

- Understand the concepts and services of cloud computing and infrastructure
- Understand the role of virtualization
- Understand the usage of programming tools in cloud computing
- Understand the security threats involved in cloud computing

Course Outline

Unit I: Introduction - evolution of cloud computing - system models for distributed and cloud computing - NIST cloud computing reference architecture - Infrastructure as a Service (IaaS) - resource virtualization - Platform as a Service (PaaS) - cloud platform & management

- Software as a Service (SaaS) - available service providers.

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

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

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

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

References:

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

CSS3E02e – NUMERICAL AND STATISTICAL METHODS

Objectives

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

Course Outcome

- Understand the basic concepts of statistics used in mathematical modelling
- Awareness on mathematical theories used in optimization

Course Outline

Unit I: 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.

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

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

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

Unit V: Introduction linear programming - mathematical formulation - graphical method of solution - simplex method - duality - dual simplex - transportation - assignment problems.

References

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

CSS4P01 – PROJECT WORK

SEMESTER IV

Objectives:

- To give a practical exposure to the process of software development life cycle.
- To develop a quality software solution by following the software engineering principles and practices. Students are also encouraged to take up a research oriented work to formulate a research problem and produce results based on its implementation/simulation/experimental analysis.

Course Outline

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

Guide has to constantly monitor the works done by the student, imparting him/her the necessary inputs for the successful completion of the project work. Students can either take up a real-life application oriented project work or research and development project. The student can formulate a project problem with the help of her/his guide and submit the project proposal of the same. Approval of the project proposal is mandatory. If approved, the student can commence working on it, and complete it.

Guidelines for Submission of Report

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

It is a test of the student's ability to undertake and complete a sustained piece of independent research and analysis / application development, and to write up the work in a coherent form according to the rules and conventions of the academic community. The role of the supervisor too is very crucial in this context.

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

The dissertation is to be prepared in TEX format (either Latex or a suitable Windows TEX

variant). The format of the report is included in Appendix A. Students are also encouraged to present their work in IT fest/conference/workshop/journal with the assistance and guidance of the supervisor. This should pave as a good start for the student in the art of publishing/presenting his/her work to the outside world. Due weightage is accommodated for publications out of the project work in the final evaluation.

CSS4E03a- Deep Learning

Objectives

- Explore the basics of deep learning and deep network architecture
- Define and train deep neural networks for real world solutions

Course Outcome

- Knowledge of deep network architectures and its application
- A good understanding of neural network models for solving real world problems that require AI based solutions

Course Outline

Unit I: Mathematical & Computational Concepts: Scalars, vectors, matrices, tensors, linear dependencies, span, norms, Eigen decomposition, SVD, Moore-Penrose Inverse. Deep learning with GPUs. Software frameworks for deep learning.

Unit II: Feed Forward Neural Network: Multilayer perceptron: input, output, hidden layers. Network training: back propagation, activations functions: linear, sigmoid, tanh, hard tanh, softmax, rectified linear. Regularization and optimization techniques.

Unit III: Deep Learning Methods: Deep learning, Convolution neural networks: Convolution & Pooling layer, Variants of the basic convolution functions, data types, Pretrained Networks, Adversarial Generative Networks, Efficient Convolution algorithm.

Unit IV: Recurrent & Recursive Neural networks: Recurrent NN, Bidirectional RNN, Deep Recurrent Networks. Recursive NN: Architecture, Unsupervised RNN, echo state Networks.

Unit V: Deep Reinforcement Learning: Reinforcement Learning, deep reinforcement learning, Deep Q Network, Double DQN, Dueling DQN.

References

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016
2. Josh Patterson, Adam Gibson, *Deep Learning: A practitioners' Approach,* O' Reily , 2017.
3. Sandro Skansi, *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence,* Springer, 2018
4. Anurag Bhardwaj, Wei Di, Jianing Wei, Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning, Packt publishing, 2018

CSS4E03b – Advanced Computing Technologies in Data Science

Objectives

- To understand about technologies for data and analysis
- To acquaint with various tools used

Course Outcome

- Develop knowledge on various technologies
- Aids in decision making for using right tools for analysis

Course Outline

Unit I: Cloud Computing: Introduction, Architecture, Deployment Models, Service Models, Cloud Security, Challenges, Importance of Data Science with Cloud Computing.

Unit II: Block Chain Technology: Introduction, Architecture, Forming a Chain, Transactions, Applications, Frameworks.

Unit III: Ecommerce: Introduction, Business Models, Payment Systems, Security Systems, Data Science for Driving Growth in ecommerce, Making sense of ecommerce, optimizing ecommerce business systems

Unit IV: Doing Data Science with Excel and Knime: Introduction to Excel, Data Analysis Using Excel, Using Knime for Advance Data Analysis.

Unit V: Case Studies: Hadoop, AWS Services: AmazonS3, Apache Flink, Apache Spark

References

1. Domenico Talia, Paolo Trunfio, Fabrizio Marozzo, Data Analysis in the Cloud: Models, Techniques and Applications
2. Isaac D. Cody, Blockchain Technology and Data Analytics. Digital Economy Financial Framework with Practical Data Analysis and Statistical Guide to Transform and Evolve Any Business
3. Lillian Pierson, Data Science For Dummies
4. Venkat Ankam, Big Data Analytics, Packt Publishing Ltd
5. Andrew Minteer, Analytics for the Internet of Things (IoT), Packt Publishing Ltd

CSS4E03c – Natural Language Processing

Objective

To get introduced to language processing technologies for processing the text data.

Course Outcome

- Understand the mathematical foundation of Natural Language Processing
- Knowledge about various models in NLP
- Familiarity about various tools and technologies used in NLP

Course Outline

Unit I: Natural Language Processing, Linguistic Background, Mathematical Foundations, Morphological Analysis, Tokenization, Stemming, Lemmatization, Boundary Determination.

Unit II: Reading unstructured data, Representing text data, Part of speech tagging, Syntactic Representation, Text similarity, WordNet based similarity, Shallow parsing, Semantic representation.

Unit III: Information retrieval and Information extraction - Named Entity Recognition - Relation Identification-Template filling.

Unit IV: Language model - Probabilistic Models - n-gram language models- Hidden Markov Model- Topic Modelling - Graph Models -Feature Selection and classifiers -Rule-based Classifiers - Maximum entropy classifier – Clustering-Word and Phrase-based Clustering.

Unit V: Tools – Natural Language Tool kit, Apache OpenNLP. Applications of Text Analytics – Applications in Social media - Life science - Legal Text–Visualization -Case studies.

References:

1. Christopher D. Manning and Hinrich Schutze, "*Foundations of Statistical Natural Language Processing*", MIT Press, 1999.
2. Steven Struhl, "*Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence*", Kogan Page, 2015.
3. Matthew A. Russell, "*Mining the Social Web*", O'Reilly Media, 2013.
4. Steven Bird, Ewan Klein and Edward Loper, "*Natural Language Processing with Python*", 1st Edition, O'Reilly Media, 2009.

CSS4E03d – COMPUTER OPTIMIZATION TECHNIQUES

Objectives:

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

Course Outcome

- Know-how of the mathematical foundations of optimization techniques
- Understanding about various optimization techniques

Course Outline

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

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

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

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

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

References:

1. H. A. Taha, *Operations Research: An Introduction*, 9th Edition, Pearson Prentice Hall, ISBN:

013255593X.

2. C. H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, ISBN: 9780486402581.

CSS4E03e – SOFT COMPUTING

Objectives

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

Course Outcome

- A good level of understanding of Soft Computing and its application
- Understand various concepts and functions used in Soft Computing

Course Outline

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

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

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

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

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

References:

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

CSS4E04a– Text Analytics

Objectives:

- To impart knowledge and skills required for text data analytics
- To train on tools and techniques needed for text data analytics
- To undertake fundamental and applied research in the domain of text data analytics with social relevance

Course Outcome

- Able to understand the currently used frameworks and methods for text analytics and natural language understanding
- Summarize the application areas, trends, and challenges in text analytics.
- Describe, review, analyze, and criticize the main text analysis methods present in scientific papers

Course Outline

UNIT I: Introduction to Natural Language Processing: Basics of NLP, Linguistics, Language Syntax and Structure, Language Semantics. Text Analytics: Introduction, Text Mining vs Text Analytics, Text Analytics Process Flow. Processing and Understanding Text: Text Pre-processing, Text tokenization, Text Normalization, Text Syntax and Structure

UNIT II: Text Classification: Basics of Text Categorization, Feature Extraction, Classification Algorithms - Naive Bayes Classifiers, Naive Bayes Variations, Support Vector Machines, Evaluating Classification Models.

UNIT III : Text Summarization: Importance of Text Summarization, Extractive and Abstractive methods for Text Summarization, Automatic Text Summarization, Examples of Text Summarization, Topic Modeling, Latent Semantic Analysis, Text Rank.

UNIT IV: Text Similarity and Clustering: Important Concepts, Feature Extraction, Analyzing Text Similarity, Analyzing Document Similarity, Text Clustering: Motivation, Document Clustering, K-Means Clustering, Affinity Propagation, Hierarchical Clustering, Evaluation of Clustering.

UNIT V: Semantic and Sentiment Analysis: Semantic Analysis, Exploring WorldNet, Word Sense Disambiguation, Named Entity Recognition, Analyzing Semantic Representations, Sentiment Analysis.

References:

1. Dipanjan Sarkar, Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data
2. Steven Struhl, Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence
3. Gary Miner, John Elder, Andrew Fast, Thomas Hill, Robert Nisbet, Dursun Delen, Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications
4. Atefeh Farzindar, Diana Inkpen, Natural Language Processing for Social Media
5. Peter Jackson, Isabelle Moulinier, Natural Language Processing for Online Applications: Text retrieval, extraction and categorization - Front Cover

CSS4E04b– IoT Analytics

Objectives:

- To teach the fundamentals of IoT and its analytics
- To understand the real-time use of cloud storage and analysis of those data.
- To understand the application of geospatial data and remote sensing.

Course Outcome

- A good understanding on IoT concepts and its working.
- Apply analytics in IoT

Course Outline

UNIT I: Introduction: IoT, IoT devices, Applications, IoT Analytics, IoT challenges

UNIT II: IoT Networking: IoT networking connectivity protocols: Bluetooth Low energy, 6LoWPAN, ZigBee, NFC, SigFox, WiFi, Cellular.IoT networking data messaging protocols: MQTT, HTTP, CoAP, DDS

UNIT III: IoT Analytics for Cloud: Building elastic analytics, Elastic analytic concepts, Designing for scale: Decouple key components, distributed computing, Cloud security and analytics, Overview on Major clouds in the industry.

UNIT IV: Geospatial Analytics to IoT Data: Basics of geospatial analysis, vector-based methods, raster-based methods, storing geospatial data, processing geospatial data.

UNIT V: Exploring IoT Data: Exploring & Visualizing data, basic time series analysis, adding internal data sets, adding external data sets, visual analysis for IoT data, dashboard with Tableau, alerts

References:

1. Andrew Minteer, Analytics for the Internet of Things (IoT), Packt Publishing Ltd, 2017
2. HwaiyuGeng, Internet of Things and Data Analytics Handbook, John Wiley & Sons, 2017
3. Gaston C. Hillar, Internet of Things with Python, Packt Publishing Ltd, 2016

CSS4E04c– Image & Video Analytics

Objectives:

- To teach the fundamentals of digital image processing, image and video analysis.
- To understand the real-time use of image and video analytics.
- To demonstrate real time image and video analytics applications and others.

Course Outcome

- Understanding of the fundamental principles of image and video analysis and have an idea of their application.
- Apply image and video analysis in real world problems.

Course Outline

UNIT I: Digital image representation, Visual Perception, Sampling and Quantization, Basic Relations between Pixels, Mathematical Tools Used in Digital Image Processing. Fundamental Operations, Vector and Matric Operations, Image Transforms (DFT, DCT, DWT, Hadamard).

UNIT II: Fundamentals of spatial filtering: spatial correlation and convolution, smoothing, blurring, Sharpening, edge detection. Basics of filtering in the frequency domain, Histograms and basic statistical models of image.

UNIT III: Colour models and Transformations, Image and Video segmentation, Image and video Demonizing, Image and Video enhancement, Image and Video compression.

UNIT IV: Object detection and recognition in image and video, Texture models Image and Video classification models, Object tracking in Video.

UNIT V: Applications and Case studies: Industrial, Retail, Transportation & Travel, Remote Sensing, Video Analytics in WSN: IoT Video Analytics Architectures.

References:

1. R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, 3rd Edition, Addison Wesley, 2007.
2. Rick Szelisk, *Computer Vision: Algorithms and Applications*, Springer 2011.
3. Jean-Yves Dufour, *Intelligent Video Surveillance Systems*, Wiley, 2013.
4. Caifeng Shan, FatihPorikli, Tao Xiang, Shaogang Gong, *Video Analytics for Business Intelligence*, Springer, 2012.
5. AsierPerallos, Unai Hernandez-Jayo, Enrique Onieva, Ignacio Julio García Zuazola, *Intelligent Transport Systems: Technologies and Applications*, Wiley,2015.
6. Basudeb Bhatta, *Analysis of Urban Growth and Sprawl from Remote Sensing Data*, Springer, 2010

CSS4E04d – Visual Data Analytics

Objectives:

- To impart knowledge and skills required for visual data analytics
- To train on tools and techniques needed for visual data analytics
- To undertake fundamental and applied research in the domain of visual data analytics with social relevance

Course Outcome

- Students will be able to understand the currently used frameworks and methods for visual analytics.
- Summarize the application areas, trends, and challenges in visual analytics

Course Outline

UNIT I : Introduction; Scene Understanding Datasets: Small-Scale Scene Understanding Datasets: 8-scene Dataset, 15-scene Dataset, UIUC Sports, CMU 300; Large-Scale Scene Understanding Datasets: 80 Million Tiny Image Dataset, PASCAL Dataset, ImageNet Dataset, LabelMe Dataset, Scene Understanding (SUN) Dataset, Places205 Dataset.

UNIT II : Indoor/Outdoor Classification with Multiple Experts: Introduction, Individual Indoor/Outdoor Experts, Individual Indoor/Outdoor Experts, Data Grouping Using Experts' Decisions, Diversity Gain of Experts Via Decisions Stacking, Expert Decision Fusion Systems, Performance Evaluation.

UNIT III : Outdoor Scene Classification Using Labeled Segments: Introduction, Review of Previous Works: Low-Level Features, Mid-Level Features, High-Level Features, Deep Features, Scene Parsing and Semantic Segmentation.

UNIT IV : Proposed Coarse Semantic Segmentation (CSS): Limitations of Traditional Learning Units, Coarse Segmentation, Segmental Semantic Labeling; Scene Classification Using CSS; Experimental Results.

UNIT V : Global-Attributes Assisted Outdoor Scene Geometric Labeling: Introduction, Review of Previous Works: Geometric Context from a Single Image, Blocks World Revisited, Single-View 3D Scene Parsing by Attributed Grammar, Inferring 3D Layout of Buildings from a Single Image; Proposed GAL System: System Overview, Initial Pixel Labeling (IPL), Global Attributes Extraction (GAE), Layout Reasoning and Label Refinement (LR2).

Reference:

1. Chen, Ren, Yuzhuo**,** Kuo**,** C.-C. Jay ,Big Visual Data Analysis - Scene Classification and Geometric, Springer Singapore ,Edition 1
2. Tamara Munzner,Visualization Analysis and Design ,2014
3. James J. Thomas,Kristin A. Cook, Illuminating The Path: The Research And Development Agenda For Visual Analytics, National Visualization and Analytics Ctr (2005)

CSS4E04e– Healthcare Data Analytics

Objectives:

- To understand about health care data and analysis
- To acquaint with various prediction models used

Course Outcome

- Develop various prediction models understanding the type of data
- Aids in decision making for diagnosis by exact prediction

Course Outline

UNIT I
Introduction to health care data analytics, Applications, Components, Data Sources: EHR, Components of EHR, other data sources. Coding systems, Benefits of HER, Barriers & Challenges, Phenotyping algorithms.

UNIT II
Biomedical Image Analysis: Biomedical image modalities, Object Detection, Image segmentation, Image Registration, Feature Extraction.

UNIT III
Biomedical Signal Analysis: Types of biomedical signals, denoising of signals: PCA, Wavelet Filtering. Multivariate Biomedical Signal. Cross correlation analysis, Recent Trends.

UNIT IV
Sensor data: Introduction to mining sensor data, applications. Genomic data: methods and standards for genomic data analysis.

UNIT V
Clinical prediction Models: Basic statistical prediction models, alternative clinical prediction models, Survival models, Evaluation & Validation.

References:

1. Chandan K. Reddy, Charu C. Aggarwal, *Health Care Data Analytics*, CRC Press, 2015
2. Trevor L Storme, *Healthcare Analytics for Quality and Performance Improvement*, John Wiley & Sons, 2013
3. Jason Burke, *Health Analytics: Gaining the Insights to Transform Health Care*, John Wiley & Sons, 2013
4. Hui Yang, Eva K. Lee, *Healthcare Analytics: From Data to Knowledge to Healthcare Improvement*, John Wiley & Sons, 2016

APPENDIX A - Guidelines for Project Report & Layout

Cover Page & First Page

<<TITLE>>

A PROJECT REPORT

SUBMITTED BY

<<NAME OF THE STUDENT>>

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

COMPUTER SCIENCE

<<COLLEGE EMBLEM>>

<<NAME OF THE DEPARTMENT>>

<<NAME OF THE INSTITUTION>>

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

MONTH YEAR

Acknowledgement

ACKNOWLEDGEMENT

I would like to thank ……..

Date: Name of the Student

Declaration by the Student

DECLARATION

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

Date: Signature:

Name: Reg. No.:

Certificate from Guide & HoD

CERTIFICATE

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

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

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

Place:

Date:

PROJECT EVALUATION REPORT OF THE EXAMINERS

Certified that the candidate was examined by us in the Project Viva Voce Examination held on ..... and his/her Register Number is .....

Examiners:

1.

2.

Contents

CONTENTS

Abstract

ABSTRACT

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

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

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

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

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

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

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

List of Figures

LIST OF FIGURES

List of Tables

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

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

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

CHAPTER 2

PROBLEM DEFINITION AND METHODOLOGY

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

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

CHAPTER 3

REQUIREMENT ANALYSIS AND SPECIFICATION

This chapter includes the following subsections.

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

CHAPTER 4

REQUIREMENT ANALYSIS AND SPECIFICATION

This chapter includes the following subsections.

1. UsersoftheSystem
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.,
  - 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.

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

CHAPTER 6
TESTING

This chapter includes the following subsections.

1. Test Plans
2. Unit Testing

a. Test Items (Test Cases)

3. Integration Testing
4. System Testing

a. Test Items (Test Cases)

5. Implementation - Changeover Plans

CHAPTER 7

CONCLUSION

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

There are a couple of rules for this chapter:

- All material presented in this chapter must have appeared already inthe 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 yourproject.
- 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 yourwork.

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

Ideas or contents taken from other sources should be properly cited. It is important that you give proper credit to all work that is not strictly your own, and that you do not violate copyright restrictions.

References should be listed in alphabetical order of authors' surname, and should give sufficient and accurate publication details. IEEE format is to be followed while preparing citations.

PUBLICATIONS OUT OF THE PROJECT WORK

A list of publications made or communicated out of the work done in the project is to be included here.

GENERAL INSTRUCTIONS

1. All chapters should contain an introduction and summary (summarizes the entire chapter content in one or two lines) sections.
2. Students have to take care that only chapters/sections relevant to their work are to be included in their report.
3. Instead of merely replicating the definitions for these sections from standard text books of Software Engineering, the student has to describe the information related to his/her work (For eg, Feasibility study should be about how the proposed work is technically/economically/operationally feasible).
4. Figures and tables are to be clear and legible.
5. Citations are to be provided wherever necessary.
6. Important code, screenshots, report formats and glossary of technical terms are to be attached as Appendices A, B, C and D respectively.

PATTERN OF QUESTION PAPER

Sl. No.	Type of Questions	Individual weightage	Total Weightage	Numberofquestions to beanswered
1	Short Answer type questions	2	2*4 = 8	4 out of 7
2	Shortessay/problem solving type	3	3*4 = 12	4 out of 7
3	Long Essay type questions	5	5*2 = 10	2 out of 4
Total			30	18

Time: 3 Hours Maximum Weightage: 30

Part A

Answer any 4 Questions. Each question carries 2 weightage (4 x 2=8 Weightage)

1
2.
3.
4.
5.
6.
7.

Part B

Answer any 4 questions. Each question carries 3 weightage (4 x 3 = 12 Weightage)

8.
9.
10.
11.
12.
13.
14.

Part C

Answer any 2 questions. Each question carries 5 weightage. (2 x 5 = 10 Weightage)

15.
16.
17.
18.

****