CURRICULUM FRAMEWORK AND SYLLABUS
FOR OUTCOME BASED EDUCATION IN
Degree of Master of Science (MSc CS) Computer Science Program
FOR THE STUDENTS ADMITTED FROM
THE ACADEMIC YEAR 2020-21 ONWARDS

Naipunnya Institute of Management and Information Technology
(Affiliated to the University of Calicut , Accredited by NAAC with B++,ISO 9001-2015 Certified)
Pongam ,Koratty East , Thrissur District ,Kerala - 680308

Table of Contents

DEPARTMENT OF COMPUTER SCIENCE

VISION

"Be the very pinnacle of academic and research excellence in Computer Applications"

MISSION

As a Department, we are committed to

- Achieve academic excellence in Computer Applications through innovative teaching and learning processes.
- To prepare the students to be professionally competent to face the challenges in the industry.
- Promote inter-disciplinary research among the faculty and the students to create state of art research facilities.
- To promote quality and ethics among the students.
- Motivate the students to acquire entrepreneurial skills to become global leaders.

Courses Offered

- BSc Computer Science
- BCA
- MSc Computer 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.

2. PROGRAMME OUTCOME (PO)

After the successful completion of the Post Graduate Programme, MSc Computer Science at University of Calicut, a student would have :

PO1	**Advanced Problem-Solving Skills**: Graduates will demonstrate the ability to identify, analyze, and solve complex, real-world problems within their field of study, applying innovative and evidence-based solutions.

PO2	**Mastery of Advanced Research Methods**: Graduates will master advanced research methodologies, including the design, execution, and dissemination of original research, contributing to the body of knowledge in their discipline.
PO3	**Professional Development and Career Readiness:** Graduates will be well-prepared for their chosen careers, possessing the necessary skills, knowledge, and experience to excel in their professional roles or pursue further advanced studies.
PO4	**Leadership and Collaborative Abilities:** Graduates will exhibit leadership qualities and the ability to work collaboratively in diverse teams, recognizing the value of teamwork, and contributing effectively to the achievement of common goals.
PO5	**Community Engagement and Service:** Graduates will engage in community service and outreach, applying their expertise to address societal issues and contribute to the betterment of the community and beyond.

3. PROGRAMME SPECIFIC OUTCOME (PSO)

PSO1	Evaluate complex real-world problems by applying principles of theoretical computing, engineering and Mathematical models.
PSO2	Modern Tool usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
PSO3	Understand all dimensions of the concepts of software application development ,projects and Aware the students to publish their work in reputed journals.
PSO4	Conceive Project Management capabilities to solve real world problems in accordance to the needs of the industry, in a specific time frame and to address the challenging requirements coming from the enterprise applications
PSO5	Design and develop computer programs/computer-based systems in the field of Computer Sciences viz. Computational Intelligence, Machine learning, Web technology, Information Retrieval Systems, Data Analytics, Communication and networking.

4. 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

i) Total Credit for Core Courses shall not be less than 60 (sixty) and not more than 68 (sixty-eight).

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

iii) 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.

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

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

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

2. 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 examinations for these courses and have to intimate /upload the results of the same to the University on the stipulated date during the III Semester. Students have to obtain only minimum pass requirements in the Audit Courses.

3. A student shall accumulate a minimum of 80 credits for the successful completion of the Programmes.

5. DURATION OF THE PROGRAMME

1. The minimum duration for completion of a four semester PG Programme is two years. The maximum period for completion is 4 years.

2. The duration of each semester shall be 90 working days, inclusive of examinations, spread over five months.

3. Odd semesters shall be held from June to October and even semesters from November to March subject the academic calendar of the University.

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 University 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 University 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 the provisions as per clause 6.4 of this regulation. 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 a University examination at the end of each semester.

2. Practical examinations shall be conducted by the University 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 University/BoS. The Question paper shall be for minimum 20 weightage and a minimum of 2 hour duration for the examination. The result has to be intimated / uploaded to the University during the Third Semester as per the notification of the University.

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.

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 University, through the college Principal.

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

(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	Pratical test	30%	3

3. Grades shall be given for the internal evaluation 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.

4. There shall be no separate minimum Grade Point for internal evaluation.

5. 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.

6. There shall not be any chance for improvement of internal marks.

7. The course teacher shall maintain the academic record of each student registered for the course, which shall be forwarded to the University, through the college Principal, after being endorsed by the Head of the Department.

8. 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.

9. 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.

10. 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.

11. 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:

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

Components	Weightage (W)	Grade Awarded	Grade Point(GP)	Weighted GP	Overall Grade of 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

Components Weightage

Components	Weightage (W)	Grade Awarded	Grade Point(GP)	Weighted GP	Overall Grade of 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	

EXTERNAL / END SEMESTER EVALUATION (ESE)

1. The semester-end examinations in theory courses shall be conducted by the University with question papers set by external experts. The evaluation of the answer scripts shall 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 University.

3. Students shall have the right to apply for revaluation or scrutiny as per rules within the time permitted for it.

4. Photocopies of the answer scripts of the external examination shall be made available to the students for scrutiny on request by them as per rules.

5. The external evaluation shall be done immediately after the examination preferably in a Centralized Valuation Camp.

5. The language of writing the examination shall be English.

6. Pattern of questions for external/ESE (theory courses):

a. Questions shall be set to assess the knowledge acquired, standard, and application 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

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	15	A+	5	5	25	
	16	-	-	-	-	
	17	-	-	-	-	
	18	B	3	5	15	
TOTAL				30	115	

7. 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**

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**	

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**

Sl.

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

SlNo	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 is 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:

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

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

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

5.SEMESTER WISE COURSE

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	2	2	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**

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**

SEMESTER I

CSS1C01 – DISCRETE MATHEMATICAL STRUCTURES

Objectives: To introduce discrete mathematics concepts necessary to understand the basic foundation of Computer Science.

Course Outcome

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

CO2: Understand allocations of set theory and operations on set and apply operations of relations and functions in discrete structures

CO3: Understand applications of Lattices and Boolean algebra in the computer science domain.

CO4: Identify Group, Ring and Field in Group Theory

CO5: Apply the concepts of graph theory and trees to formulate problem solving and understand applications of Graph Theory and Tree

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**3**	**2**	**1**	**2**	**0**	**3**	**2**	**3**	**0**	**2**
CO2	**2**	**2**	**1**	**2**	**0**	**3**	**1**	**1**	**0**	**2**
CO3	**1**	**0**	**0**	**1**	**1**	**3**	**2**	**2**	**1**	**2**
CO4	**1**	**2**	**1**	**0**	**0**	**2**	**1**	**2**	**0**	**2**
CO5	**3**	**2**	**1**	**1**	**0**	**3**	**2**	**1**	**0**	**3**

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 CutSets, 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

CO1: Summarize different categories of data structures and design algorithms to perform operations with linear and non – linear data structures.
CO2: Describe how arrays, linked lists, stacks, queues, trees and graphs are represented in memory and used by algorithms.

CO3: Describe common applications for arrays, linked lists, stack, queue, tree and graphs.
CO4: Demonstrate different methods for traversing trees and describe various types of trees and heap structures.
CO5: Design and implement an appropriate hashing function for an application and Discuss the computational efficiency of the principal algorithms for sorting, searching and hashing.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	**0**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**
CO2	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**0**	**0**	**3**
CO3	**0**	**0**	**0**	**3**	**1**	**3**	**0**	**0**	**0**	**0**
CO4	**0**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**
CO5	**2**	**0**	**0**	**3**	**0**	**2**	**0**	**0**	**0**	**0**

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

Unit III:

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

Unit IV:

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

Unit V:

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

References:

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley, ISBN: 978-0201000238.

2. Horowitz E and Sahni S, Fundamentals of Data Structures, Computer Science Press, ISBN: 9780716780427.

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

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

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

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

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

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

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

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

CSS1C03 – THEORY OF COMPUTATION

Objectives:

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

Course Outcome

CO1: Describe broad overview of the theoretical foundations of computer science.
CO2: Understand regular languages and finite automata.
CO3: Apply the concept of context free languages in problem solving.
CO4: Solve various problems of applying normal form techniques, push down automata and Turing Machines.
CO5: Propose solutions for the problems based on computability and decidability.

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**4**	**1**	**1**	**3**	**3**	**2**	**3**	**3**	**3**	**3**
CO2	**3**	**3**	**3**	**1**	**3**	**2**	**4**	**4**	**3**	**2**
CO3	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**4**	**2**
CO4	**2**	**3**	**2**	**4**	**2**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**3**	**3**	**4**	**4**	**2**	**3**	**3**	**3**	**3**

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 simulationsClasses 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.

CSS1C04 – THE ART OF PROGRAMMING METHODOLOGY

Objectives:

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

Course Outcome

CO1: Improve ability to develop effective algorithms.
CO2: Understand the fundamental principles of problem-solving using computers.

CO3. Demonstrate the applications of the programming constructs including decision making, looping, arrays and strings and Conceptualize modular programming basics using functions, structures and Unions.
CO4. Understand features like pointers and macros and to become familiar with programming with files
CO5: Design, develop, implement, test and document well-structured and reliable computer programs using the C programming language.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO2	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO3	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

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:

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

Unit III:

Decision making - if statement, if else statement, nesting of if else and else if ladder, switch statement, break statement, continue statement, goto statement, return statement. Looping - while, do-while, and for loops, nesting of loops, skipping & breaking loops. Arrays – single dimension arrays - accessing array elements - initializing an array, two dimensional & multidimensional arrays - memory representation - strings - processing of strings – string manipulation functions.

Unit IV:

The Concept of modularization - defining function - types of functions - User defined functions - function prototype and definition - arguments - passing parameters - call by reference - call by value - returning - nesting of functions and recursion - passing arrays & strings to function - returning multiple values - recursion - scope and life time of variables storage class specifiers - automatic, extern, static storage, register storage. Structures & Union definition, giving values to members, structure initialization, comparison of structure variables, arrays of structures, arrays within structures, structures within arrays, structures and functions, Unions, bit-fields.

Unit V:

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

References:

1. Martin M. Lipschutz and Seymour Lipschutz, Schaum's Outline of Theory and Problems of Data Processing, ISBN: 9780070379831 (Unit I Part A).

2. Anil Bikas Chaudhuri, The Art Of Programming Through Flowcharts & Algorithms, Laxmi Publications, New Delhi (Unit I Part A).

3. Jean Paul Trembley and Pual G Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill (Unit I Part B).

4. R G Dromey, How to Solve by Computer, Pearson Education, 5th Edition, ISBN: 0134340019 (Unit I Part B).

5. J.B Dixit, Computer Fundamentals and Programming in C, Firewall Media, ISBN: 8170088828. (Unit I Part C).

6. Dennie Van Tassel, Program Style, Design, Efficiency, Debugging, and Testing, PHI, ISBN: 0137299478 (Unit I Part C).

7. E Balagruswamy, Programming in ANSI C, TMH, 5th Edition, ISBN: 0070681821.

8. Kamthane, Programming in C, 2nd Edition, Pearson India, ISBN: 8131760316.

9. Brian W. Kernighan and Dennis M. Ritchie, C Programming Language, PHI, ISBN: 0131103628.

10. Kanetkar, Let Us C, BPB Publications, 8th Edition, ISBN: 1934015253.

CSS1C05 – COMPUTER ORGANIZATION & ARCHITECTURE

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

Course Outcome

CO1: Identify, understand and apply different number systems and codes and identify the digital representation of data in a computer system.
CO2: Understand the general concepts in digital logic design and their use in sequential and combinational circuit design.
CO3: Describe fundamental organization of a computer system and computer arithmetic formulae and solve problems .
CO4: Explain addressing modes, instruction formats and program control statements.

CO5: Distinguish the organization of various parts of a system memory hierarchy Identify and compare different methods for computer I/O.

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	0	3	3	3	3	3	0	3
CO2	3	1	0	3	3	3	3	3	0	3
CO3	3	1	0	3	3	3	3	3	0	3
CO4	3	1	0	3	3	3	3	3	0	3
CO5	3	1	0	3	3	3	3	3	0	3

Course Outline

Unit I:

Number systems and Conversions, Boolean Algebra - Truth Tables - Logic gates and Map simplification - flip-flops - design of combinational and sequential circuits - examples of digital circuits - adders, multiplexers, decoders, counters, shift registers - register transfer language and micro operations - data representation - data types, sign and magnitude, complements, fixed-point representation, floating-point representation, other binary codes, error detection codes.

Unit II:

Basic computer organization - machine instructions - classification, function, addresses, size, addressing modes - instruction cycle - instruction sequencing. Fundamental concepts - registers, register transfers, performing arithmetic or logic operations, memory read and write, execution of a complete instruction, branch instruction, single bus, two bus, three bus organization, a complete processor - Control unit - hardwired control, micro programmed control, micro instructions-types.

Unit III:

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

Unit IV:

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 ,devices - programmed I/O, interrupt I/O - interrupts – interrupt processing - hardware interrupts - programmable interrupt controller - vectored interrupts - interrupt nesting - daisy chaining - direct memory access (DMA) - DMA operations & DMA Controller, Introduction to I/O interfaces, I/O channels, IO Processors.

Unit V:

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, McGraw Hill International Edition, 5th Edition, ISBN: 9780071122184.

2. Morris Mano, Digital Logic and Computer Design, Prentice Hall of India, ISBN: 0876924178.

3. M Morris Mano, Computer System Architecture, Prentice Hall, 3rd Edition. ISBN: 0131755633.

4. William Stallings, Computer Organization and Architecture, 9th Edition, Prentice Hall, ISBN: 013293633X.

5. Andrew S Tanenbaum, Structured Computer Organization, Prentice Hall, 6th Edition, ISBN: 0132916525.

6. Floyd Thomas L, Digital Fundamentals, Pearson Education, 10th Edition, Prentice Hall, ISBN: 0132359235.

7. Albert Paul Malvino, Donald P Leach, Digital Principles and Applications, McGraw Hill, 4th Edition, ISBN: 0070398836.

8. Thomas C Bartee, Digital Computer Fundamentals, McGraw Hill, 6th Edition, ASIN: B004H0SL5K.

9. Ramesh. S. Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, 6th Edition, Wiley Eastern Ltd, New Delhi, ISBN: 9788187972884.

10. Mohamed Rafiquzzaman, Introduction to Microprocessors and Microcomputer Based System Design, 2nd Edition, CRC Press, ISBN: 9780849344756.

11. Muhammad Ali Mazidi, Janice Mazidi, Rolin Mckinlay, Janice M. Mazidi, Janice Gillispie Mazidi and Rolin D., The 8051 Microcontroller and Embedded Systems, Pearson Education Asia, 5th Indian Reprint, ISBN: 013119402X.

CSS1L01 – PRACTICAL I

Objectives: To practically implement the theory portions covered in The Art of Programming Methodology (CSS1C04) and Advanced Data Structures (CSS1C02).

Course Outcome

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

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

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

CO4: Implement files handling and command line arguments.

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

CO6: Demonstrate the concept of tree traversal and its operations.

CO7: Design program based on the concepts of sorting and searching techniques.

Course Outline

Unit I: C Programming

1. Simple C Programs like area of a circle, checking whether a given number is odd or even.

2. Implementation of programs using Loops (pyramid printing, factorial computation, number reversing, checking for Armstrong numbers, finding first N or Nth Prime numbers etc).

3. Use of 1D and 2D Arrays (searching, sorting and vector operations, matrix addition, matrix multiplication).

4. String Manipulations.

5. Structures and Unions (like addition of two complex numbers, student record creation and manipulation etc).

6. Writing functions.

7. Implementation of recursion (recursive function to compute a factorial, reverse string

etc).

8. Command line arguments.

9. Pointers - simple programs to learn concept of pointers, array operation using pointers etc.

10. File operations – file and structures.

Unit 2: Data Structures and Algorithms

1. Implementation of stacks using arrays.

2. Implementation of queues, circular queue using arrays.

3. Implementation of sequential search and binary search techniques.

4. Implementation of linked lists and operations (add, insert, delete, search) on linked lists.

5. Implementation of stacks using linked list.

6. Implementation of queues using linked list.

7. Implementation of doubly linked list.

8. Implementation of circular linked list.

9. Implementation of binary tree and traversals.

10. Implementation of Binary search trees and perform the operations on BST.

11. Implementation of various sorting algorithms.

12. Conversion of an infix expression to the postfix form using stacks.

13. Evaluation of a postfix expression.

14. Implementation of graphs and graph traversals.

15. Implementation of heap tree and operations.

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.

Course Outcome

CO1: Understand research terminology.
CO2: Apply the ethical principles of research.
CO3: Identify the components of a literature review process.
CO4: Critically analyze published research works.
CO5: Innovate and apply research methods in the discipline of computing.

Course Evaluation & Course Credit

The Ability Enhancement Audit Course has 4 credits which will not be counted for evaluating the overall SGPA & CGPA. The College/Department shall conduct examination of 2 Hrs duration with a minimum of 20 weightage before the conclusion of first semester classes and have to intimate /upload the results of the same to the University on the stipulated date during the III Semester. Students have to obtain only minimum pass requirements in this Audit Course.

Course Delivery Mode

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

Course 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 an 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

CO1: Design algorithms in context of space and time complexity and apply asymptotic notation.
CO2: Analyze the problem and develop the algorithms related to these problems.
CO3: Classify the problems and apply the appropriate design strategy to develop algorithms. CO4: Analyze the problem and develop the algorithms related to these problems.
CO4: Analyze the problem and develop the algorithms related to these problems.
CO5: Demonstrate the use of parallel algorithms.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**

CO1	**3**	**2**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**3**
CO2	**3**	**2**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**3**
CO3	**3**	**2**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**2**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**2**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**3**

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, Deterministic symmetry breaking.

References:

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

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

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

Course Outcome

CO1: Understand the basic components of a computer operating system and interpret the applications of Process and threads. .

CO2: Describe the policies for scheduling, deadlocks, synchronization, system calls, and file systems.
CO3: Illustrate the functioning of process management, memory management and file management Modules present in an OS.
CO4: Differentiate various types of scheduling algorithms.
CO5: Understand the concepts of Three-Tier Client/Server Architecture, Middleware and the characteristics of mobile operating systems.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	**1**	**0**	**0**	**3**	**0**	**3**	**3**	**3**	**0**	**1**
CO2	**1**	**3**	**0**	**3**	**0**	**3**	**3**	**3**	**0**	**2**
CO3	**2**	**3**	**0**	**3**	**0**	**3**	**3**	**3**	**0**	**2**
CO4	**3**	**3**	**0**	**3**	**0**	**3**	**3**	**3**	**0**	**3**
CO5	**1**	**3**	**0**	**3**	**0**	**3**	**3**	**3**	**0**	**3**

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.

References

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 McGrawHill 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 backdrop of Internet protocol stack.

Course Outcome

CO1: Understand the basic concepts of computer network organization and implementation.
CO2: Describe theoretical understanding of layered network models - OSI and TCP/IP Models.
CO3: Illustrate the functionalities of different network layers.
CO4: Analyze the network application such as data transmission between client and server, file transfer, real-time and multimedia transmission.
CO5: Explain the security aspects in networks and principles of cryptography.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	**2**	**0**	**0**	**3**	**3**	**1**	**0**	**2**	**0**	**1**
CO2	**1**	**0**	**0**	**3**	**3**	**3**	**2**	**1**	**1**	**2**
CO3	**2**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

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: 97898123598652000.

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

Objectives:

To introduce concepts of Artificial Intelligence and Machine Learning.

Course Outcome:

CO1: Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems in the analysis and design of information systems.
CO2: Conceptualize various knowledge representation techniques.
CO3: Analyze the problem-solving methods and algorithms related to searching, reasoning, game playing and machine learning.
CO4: Understand the functioning of expert systems and its importance.

CO5: Demonstrate the implementation of various AI algorithms to solve real life problems. Course Outline

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**3**	**2**	**4**	**4**	**3**	**3**	**3**	**3**	**3**	**3**
CO2	**3**	**2**	**3**	**4**	**3**	**3**	**3**	**3**	**3**	**3**
CO3	**2**	**2**	**3**	**4**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**2**	**3**	**4**	**3**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**2**	**3**	**4**	**3**	**3**	**3**	**3**	**3**	**3**

Unit I: Introduction - Artificial Intelligence - problems, scope and applications, problem space and search - production system- characteristics - the predicate calculus, inference rules, structures and strategies for state space search, strategies for space search, using state space to represent reasoning with the predicate calculus.

Unit II: Heuristics Search: control and implementation of state space search, generate and test, hill climbing, Best-first search, problem reduction, constraint satisfaction, means- ends analysis, heuristic in games, complexity issues.

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

Unit IV: Game playing - the mini-max search procedure, adding alpha-beta cut-offs, additional refinement, iterative deepening, planning system and its components, understanding, understanding as constrained satisfaction. Slot and filler structures: semantic nets, frames, conceptual dependency, scripts. Definition and characteristics of expert system, representing and

using domain knowledge, expert system shells. Knowledge engineering, knowledge acquisition, expert system life cycle & expert system tools, MYCIN & DENDRAL examples of expert system.

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

References:

1. Elaine Rich, Kevin Knight and Shivshankar B. Nair, Artificial Intelligence, 3rd Edition, Tata - McGraw Hill, New Delhi, ISBN: 0070087709.

2. V S Janakiraman, K Sarukesi and P Gopalakrishnan, Foundations of Artificial Intelligence and Expert System, Macmillan India Limited, ISBN: 0333926250.

3. Stuart Russell and Peter 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

CO1: Understand the software process and development models.
CO2: Understand the software design process and structured analysis of systems.
CO3: Distinguish different types of modelling like DFD and UML.

CO4: Illustrate the knowledge about the design of user interface.
CO5: Apply the skill of project management and report preparation.

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO2	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO3	**3**	**1**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

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 the theory portions covered in the courses Operating System Concepts (CSS2C07) and Computer Networks (CSS2C08) and to extend the programming knowledge acquired through course

The Art of Programming Methodology (CSS1C04).

Course Outcome

CO1: Discuss and formulate the problems based on the basic principles of networks.

CO2: Implementation of different memory management techniques in OS.

CO3: Implement various system operations of the operating system and also the various process scheduling algorithms.

CO4: Understand the TCP/IP configuration for Windows and Linux.

CO5: Design and implement various network applications such as data transmission between client and server, file transfer, real-time multimedia transmission.

CO6: Understand different Linux/UNIX shell scripts and execute various shell programs.

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)*.

Unit II: Operating System Concepts

1. Write programs using the following system calls: fork(), execl() and wait().

2. Write File System Calls to write, append and display.

3. To accept the burst time for a set of processes for FCFS scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.

4. To accept the burst time for a set of processes for SJF scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.

5. To accept the burst time and priority for a set of processes for Priority scheduling and create chart consisting of the burst time, priority, turnaround time and wait time of each process.

6. To create n Fibonacci numbers and prepare a list of prime numbers amongst them (use pipe for IPC).

7. To demonstrate IPC using shared memory.

8. To allocate memory requirements for processes using best fit allocation- Accept n processes with their memory requirements and n holes with their sizes. Perform memory allocation using Best Fit algorithm. Display a chart consisting of the process and the allocated hole.

9. To accept n processes with their memory requirements and n holes with their sizes. Perform memory allocation using First Fit algorithm. Display a chart consisting of the process and the allocated hole.

10. To demonstrate the process of contiguous allocation of memory blocks to store files of varying sizes.

11. To implement Producer Consumer problem using semaphores.

CSS2A02 – TERM PAPER (PROFESSIONAL COMPETENCY AUDIT COURSE)

Objectives:

▪ To introduce the student to the techniques of literature survey.

▪ To acquaint him/her with the process of presenting his/her work through seminars and technical reports.

Course Outcome:

CO1: Apply critical thinking skills analytical ability in problem solving.
CO2: Apply foundational research skills to address research problem.
CO3: Innovate, experiment and analyze research findings.
CO4: Demonstrate capacity to lead and manage change through a collaborative environment.
CO5: Innovate, experiment and analyze research findings and practice the process of scientific publishing.

Course Outline

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

ideas/designs/algorithms/theoretical contributions in the form of theorems and proofs/new methods of proof/new techniques or heuristics with analytical studies/implementations and analysis of results. The student should give a seminar on his/her work, during the semester, and submit a technical report. Technical report should be prepared in TEX in IEEE conference style format. Course Delivery Mode Students are 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.

Reference: 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

CO1: Explain the basics of database management systems, concepts of relational data model, entity-relationship model, relational database design, relational algebra and calculus.
CO2: Apply the normalization techniques to improve the database design.
CO3: Describe various database manipulation commands in SQL.
CO4: Understand Transaction Processing & Locking using the concept of Concurrency control.
CO5: Conceptualize advanced features of Object-Oriented Database Management Systems and Distributed databases.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	**1**	**1**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**0**
CO2	**3**	**1**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**0**
CO3	**3**	**1**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**0**
CO4	**3**	**1**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**0**
CO5	**3**	**1**	**0**	**3**	**0**	**3**	**3**	**3**	**3**	**0**

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 d&tg9,mg9lei. 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, modifying the structure of the tables, renaming, dropping of tables. Data constraints - I/O constraints, primary key, foreign key, unique key constraints, ALTER TABLE command database manipulation in SQL - computations done on table data - SELECT command, logical operators, range searching, pattern matching, grouping data from tables in SQL, GROUP BY, HAVING clauses. Joins - joining multiple tables, joining a table to it. DELETE - UPDATE. Views - creation, renaming the column of a view, destroys view. Program with SQL - data types Using SET and SELECT commands, procedural flow, IF, IF /ELSE, WHILE, GOTO, global variables. 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:

CO1: Recall the object-oriented programing concepts and basics of Java.
CO2: Design and implement object-oriented programs including packages and interfaces.
CO3: Explain and handle exceptions and threads.
CO4: Develop interactive programs using applets, AWT and swings.
CO5: Explain the concepts of JDBC, sockets and gives an introduction to Unified Modelling Language (UML).

PO→	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5

CO ↓										
CO1	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**
CO2	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**2**	**0**	**3**
CO3	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**2**	**0**	**3**
CO4	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**
CO5	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**

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 in a Nutshell: A Desktop Quick Reference, 3rd Edition, O'Reilly Media, ISBN: 0596101422.

5. Grady Booch, James Rumbaugh and Ivar Jacobson, The Unified Modeling Language User Guide, 2nd Edition, Pearson, ISBN: 8131715825.

CSS3C13 – PRINCIPLES OF COMPILERS

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

Course Outcome

CO1: Understand the major phases of compilation, identify tokens of a typical high -level programming language, define regular expressions for tokens, design and implement a lexical analyzer.
CO2: Develop the parsers and experiment with the knowledge of different parsers design without automated tools.
CO3: Construct the intermediate code representations and generation.
CO4: Explain the role of different types of runtime environments and memory organization for implementation of typical programming languages.

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

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**2**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**1**	**3**
CO2	**2**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO3	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

Course Outline

Unit I: 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: 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: Intermediate code generation – DAG – three address code – addresses and instructions – quadruples – triples – Static Simple Assignment form – types and declarations-type expressions - type equivalences – declarations – type checking – rules – type conversion - 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: 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: 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 the theoretical aspects covered in Advanced Database Management System (CSS3C11) and Object-Oriented Programming Concepts (CSS3C12) and to extend the programming knowledge acquired through The Art of Programming Methodology (CSS1C04) to encompass object-oriented techniques.

Course Outcome

CO1: Design and development of relational database systems.

CO2: Understand various advanced queries execution such as relational constraints, joins, set operations, aggregate functions, trigger and views.

CO3: Apply various software to design and build ER Diagrams, UML, Flowchart for related database systems.

CO4: Design and implement database applications on their own.

CO5: Apply JDBC to provide a program level interface for communicating with database using Java programming.

CO6: Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs.

CO7: Understand Java programming concepts and utilize Java Graphical User Interface in program writing.

CO8: Design and develop Java programs that solve real-world problems.

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 – COMPUTER GRAPHICS

Objectives:

▪ To understand the fundamentals of the modern computer graphics.

▪ To pipeline the mathematics of affine transformations in three dimensions.

▪ To understand the common data structures to represent and manipulate geometry, colour and light representation and manipulation in graphics systems.

▪ To have an exposure to programming in Open GL.

Course Outcome

CO1: Understand the basics of computer graphics, different graphics systems and applications of computer graphics.

CO2: Extract scene with different clipping methods and its transformation to graphics display device.
CO3: Explore projections and visible surface detection techniques for display.
CO4: Explore object representations and surface detection methods.
CO5: Understand techniques and OpenGL programming concepts.

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO2	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO3	**3**	**3**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**3**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**3**	**0**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

Course Outline

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

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

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

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

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

References:

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

2. Donald D. Hearn, M. Pauline Baker and Warren Carithers, Computer Graphics with Open GL, 4th Edition, Prentice Hall, ISBN: 9780136053583.

3. Hill, Computer Graphics using OpenGL, 3rd Edition, Prentice Hall of India Private Ltd. New Delhi, ISBN: 8120338294.

4. Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, Dave Shriner and Tom David, Open GL Programming Guide, 6th Edition, Person, ISBN: 9780201604580.

5. The Official Guide to Learning OpenGL, Version 1.1, Available at http://www.glprogramming.com/red/.

6. Shreiner and Angel, Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, 6th Edition, Pearson Education, ISBN: 0132545233.

CSS3E02c – CRYPTOGRAPHY AND NETWORK SECURITY

Objectives:

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

▪ To understand various aspects of network security standards.

Course Outcome

CO1: Understand the fundamentals of cryptography.
CO2: Describe data integrity, authentication, digital signatures.

CO3: Analyze different network security applications
CO4: Familiarize standard algorithms that provide confidentiality, integrity and authenticity.
CO5: Understand network security technologies.

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**3**	**0**	**4**	**3**	**2**	**3**	**3**	**1**	**3**	**3**
CO2	**3**	**1**	**3**	**3**	**2**	**3**	**3**	**2**	**3**	**3**
CO3	**3**	**1**	**3**	**3**	**2**	**3**	**3**	**3**	**3**	**4**
CO4	**3**	**1**	**3**	**4**	**2**	**3**	**3**	**3**	**3**	**3**
CO5	**3**	**1**	**3**	**4**	**2**	**3**	**3**	**3**	**3**	**3**

Course Outline

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

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

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

Unit IV: Transport level security - web security considerations - secure socket layer and transport layer security - SSL architecture - SSL record protocol - change cipher spec protocol - handshake protocol. Transport layer security - HTTPS - SSH. IP Security - overview - policy - encapsulating security payload - combining security associations - internet key exchange.

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

References:

1. William Stallings, Network Security Essentials Applications and Standards, 4th Edition, Pearson India, ISBN: 8131761754.

2. William Stallings, Cryptography and Network Security: Principles and Practice, 6th Edition, Pearson India, ISBN: 9332518777.

3. Atul Kahate, Cryptography and Network Security, 3rd Edition, Tata McGraw- Hill Publishing, ISBN: 9789332900929.

4. Eric Maiwald, Fundamental of Network Security, 1st Edition, Tata McGraw - Hill Education, 0071070931.

5. Charlie Kaufman, Radia Perlman and Mike Speciner, Network Security: Private Communication in Public WorJd, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120322134.

SEMESTER IV

CSS4P01 – PROJECT WORK

Objectives:

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

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

Course Outcome:

CO1: Demonstrate a depth of knowledge of modern technology.
CO2: Practice to communicate effectively and to present ideas clearly and coherently to specific audiences in both the written and oral forms.

CO3: Understand the project requirements, reflect on their learning and take appropriate actions to implement it.
CO4: Estimate, plan, calculate, and adjust project variables.
CO5: Understand the importance of iteration, evaluation and prototyping in design of a software system.

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 real-life application-oriented project work or research and development projects. The student can formulate a project problem with the help of her/his guide and submit the project proposal of the same. Approval of the project proposal is mandatory. If approved, the student can commence working on it, and complete it. Guidelines for Submission of Report The distinguishing mark of a dissertation is an original contribution to knowledge. The dissertation is a formal document whose sole purpose is to prove that you have made an original contribution to knowledge. Failure to prove that you have made such a contribution generally leads to failure. It is a test of the student's ability to undertake and complete a sustained piece of independent research and analysis / application development, and to write up the work in a coherent form according to the rules and conventions of the academic community. The role of the supervisor too is very crucial in this context.

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

CSS4E03f – WEB ENGINEERING

Objectives:

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

Course Outcome

CO1: Understand basic concepts Web engineering
CO2: Describe Requirements Engineering (RE) for web applications and familiarize Web application architecture and architecture for multimedia data.
CO3: Understand NOSQL databases and introduce MongoDB
CO4: Understand the basics of Modelling web applications and web application design.
CO5: Understand the elementary concepts of testing web applications.

PO→ / **CO ↓**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	**3**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**
CO2	**3**	**3**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**
CO3	**1**	**0**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**
CO4	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**
CO5	**3**	**1**	**0**	**3**	**3**	**3**	**3**	**3**	**0**	**3**

Course Outline

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

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

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

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

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

References

1. Gerti Kappel, Birgit Proll, Siegried Reich and Werner Retschitzegger, Web Engineering: The Discipline of Systematic Development of Web Applications, John Wiley and Sons Ltd, ISBN: 9780470064894.

2. Roger S Pressman and David Lowe, Web Engineering: A Practitioner's Approach, 1st Edition, Tata Macgraw Hill Publications, ISBN: 9780073523293.

3. Leon Shklar and Rich Rosen, Web Application Architecture: Principles, Protocols and Practices, 2nd Edition, Wiley, ISBN: 047051860X.

4. Guy W Leeky-Thompson, Just Enough Web Programming with XHTML, PHP, and MySQL, 1st Edition, Cenagage Learning, ISBN: 159863481X.

5. Anders Moller and Michael Schwartzbach, An Introduction to XML and Web Technologies, 1st Edition, Pearson Education, New Delhi, 2009.

6. Chrits Bates, Web Programming: Building Internet Applications, 3rd Edition, Wiley India Edition, ISBN: 8126512903. MySQL, 1st Edition, Cenagage Learning, ISBN: 159863481X.

CSS4E04a – DIGITAL IMAGE PROCESSING

Objectives:

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

Course Outcome

CO1: Understand the fundamental concepts of a digital image processing
CO2: Apply various image enhancement techniques
CO3: Describe various image enhancement techniques
CO4: Implement algorithms for handling intensive image restoration problems.
CO5: Identify and compare various image segmentation and representation techniques and understand various image compression procedures.

PO→ / CO ↓	PO1	PO2	PO3	PO4	PO5	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO2	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO3	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
CO4	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

CO5	**3**	**1**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

Course Outline

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

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

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

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

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

References

1. Richard E Woods and Rafael C Gonzalez, Digital Image Processing, 3rd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131726959.

2. B. Chanda and D.D. Majumder, Digital Image Processing and Analysis, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120343255.

3. A.K. Jain, Fundamentals of Digital Image Processing, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120309294.

4. W.K. Pratt, Digital Image Processing: PIKS Scientific Inside, 4th Edition, John Wiley, ISBN: 0471767778.

5. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing Analysis and Machine Vision, 3rd Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131518833.