VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
(Autonomous)
Kanuru, Vijayawada – 520 007
(Approved by AICTE, Accredited by NBA, and ISO 9001: 2008 Certified)
(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

Academic Regulations for M.Tech(VR10) w.e.f: 2010-2011
(Common to all branches)

1. INTRODUCTION

Academic Programmes of the College are governed by rules and regulations as approved by the Academic Council, which is the highest Academic body of the Institute. These academic rules and regulations are effective from the academic year 2010-11, for students admitted into two year PG programme offered by the college leading to Master of Technology (M.Tech) in various specializations offered by respective departments as given in Table 1.

2. PROGRAMMES OFFERED

Presently, the college is offering Post Graduate programmes in Engineering with the following specializations:

Table 1: List of Specilizations

S.No	Specialization	Department
1	Structural Engineering	Civil Engineering
2	Computer Science and Engineering	Computer Science and Engineering
3	Communication s and Signal Processing	Electronics & Communication Engineering
4	Telematics	
5	Power Systems	Electrical & Electronics Engineering
6	CAD/CAM	Mechanical Engineering
7	Thermal Engineering	

3. DURATION OF THE PROGRAMME

The duration of the programme is two academic years consisting of four semesters. A student is permitted to complete the postgraduate programme in a stipulated time frame of 4 years from the date of joining. Otherwise he/she shall forfeit their seat in M.Tech Programme and the admission shall stand cancelled.

4. MINIMUM INSTRUCTION DAYS

Each semester, normally consists of a minimum of 90 instruction days with about 30 to 35 contact periods per week.

5. ELIGIBILITY CRITERIA FOR ADMISSION

The eligibility criteria for admission into M.Tech programme are as per the guidelines of APSCHE .

5.1 CATEGORY –A Seats:

- These seats will be filled by the Convener, PGCET Admissions.

5.2 CATEGORY –B Seats :

- These seats will be filled by the College as per the guidelines of APSCHE

6. PROGRAMME STRUCTURE

Every specialization of the M.Tech programme shall have six theory courses and two practical / mini project / seminar courses in each of first and second semesters. A major project is offered in third and fourth semesters.

6.1 Course Code and Course Numbering Scheme

Course Code consists of eight characters in which the first four are alphabets and rest are numerals. First four characters are described in Table 2 and 3.

Table 2: First and Second Character description

First Two Characters	Name of the Department
CE	Civil Engineering Department
CS	Computer Science and Engineering Department
EC	Electronics & Communication Engineering Department
EE	Electrical & Electronics Engineering Department
ME	Mechanical Engineering Department

Third and fourth character represents specialization offering as mentioned in Table No. 3.

Table 3: Third and Fourth Character description

Next Two Characters	Name of the Specialization
SE	Structural Engineering
CS	Computer Science and Engineering
SP	Communication s and Signal Processing
TM	Telematics
PS	Power Systems Engg
CC	CAD/CAM
TE	Thermal Engineering

Fifth and sixth characters represent semester number and syllabus version number of the course offered Seventh character represents course type, as per Table No. 4

Table 4: Course type description

SEVENTH CHARACTER	DESCRIPTION
0	Theory course
5	Lab course

Eighth character represents course number as described in Figure 1 below. However, few courses are given distinct codes.

For example, in **MECC 1051** course, the course is offered by Mechanical Engineering Department (**ME**) in CAD/CAM specialization offered in the first semester (**1**), the course syllabus version number (**0**), the course is of lab type (**5**) and the course number is (**1**), as given in figure.2 below.

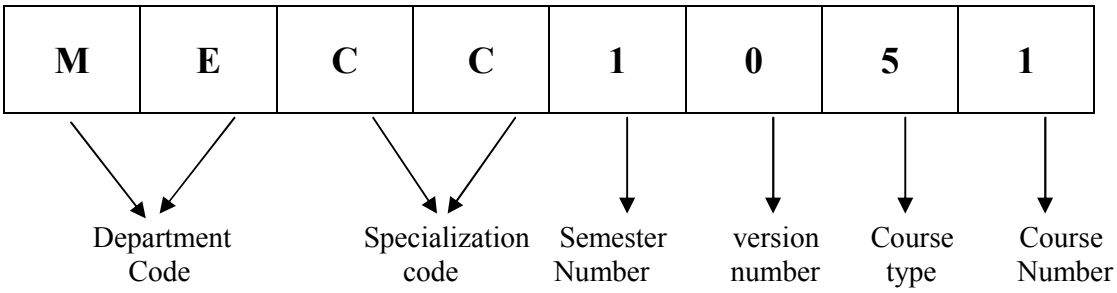

Figure 1: Course Code Description

6.2 Scheme of Instruction for 1st and 2nd Years

- The scheme of instruction and exact syllabi of all postgraduate programmes are given separately.

6.3 Contact Hours and Credits

The Course Credits are broadly fixed based on the following norms:

- Lectures – One Lecture period per week is assigned one credit.
- Tutorials - Two tutorial periods per week are assigned one credit.
- Practical – 2 periods per week is assigned one credit
- Seminar/Mini Project shall have 2 credits.
- Major project shall have 24 credits.
- However, some courses are prescribed with fixed number of credits depending on the subject complexity and importance.

6.4 Theory / Tutorial Classes

Each course is prescribed with fixed number of lecture periods per week. During lecture periods, the course instructor shall deal with the concepts of the course. For certain courses, tutorial periods are prescribed, to give exercises to the students and to closely monitor their learning ability.

6.5 Laboratory Courses

A minimum prescribed number of experiments have to be performed by the students, who shall complete these in all respects and get each experiment evaluated by teacher concerned and certified by the Head of the Department concerned at the end of the semester.

6.6 Programme Credits

Each specialization of M.Tech programme is designed to have a total of 80 credits, and the student shall have to complete the courses and earn credits as per the requirements for the award of degree.

7. MEDIUM OF INSTRUCTION

The medium of instruction and examination is English.

8. SYLLABUS

As approved by the concerned BOS and the Academic Council.

9. ELIGIBILITY REQUIREMENT FOR APPEARING SEMESTER END EXAMINATION AND CONDONATION

a. Regular course of study means a minimum average attendance of 75% in all the courses computed by totaling the number of periods of lectures, tutorials, practical courses and project work as the case may be, held in every course as the denominator and the total number of periods attended by the student in all the courses put together as the numerator.

b. Condonation of shortage in attendance may be recommended by respective Heads of Departments on genuine medical grounds, provided the student puts in at least 65% attendance in each subject and provided the Principal is satisfied with the genuineness of the reasons and the conduct of the student.

c. Students, having shortage of attendance, shall pay Rs.20/-per every period of attendance shortage subject to a minimum of Rs.500/-.

d. Minimum of 50% aggregate marks must be secured by the candidates in the internal examinations conducted for theory, practice and lab courses, to be eligible to write semester end examinations. However, if the student is eligible for promotion based on the attendance, in case necessary, a shortage of internal marks up to a maximum of 10% may be condoned by the Principal based on the recommendations of the Heads of the Departments.

e. Students having shortage of internal marks up to a maximum of 10% shall have to pay Rs.1000/- towards condonation fee for shortage of internal marks.

f. A student, who does not satisfy the attendance and/or internal marks requirement, shall have to repeat that semester.

g. Eligible candidates who failed to register for all papers for the semester-end examinations shall not be permitted to continue the subsequent semester and has to repeat the semester for which he/she has not registered for semester end examinations.

10. EXAMINATIONS AND SCHEME OF EVALUATION

10.1Internal Examinations:

10.1.1 Theory Courses

Each course is evaluated for **40** marks (a+b)

a) The internal evaluation shall be made based on the two mid term examinations each for 2**0** marks will be conducted in every theory course in a semester. The mid term marks shall be awarded giving a weightage of $2/3^{rd}$ in the examination in which the student scores more marks and $1/3^{rd}$ for the examination in which the student scores less marks. Each midterm examination shall be conducted for duration of 90 minutes with 3 questions to be answered out of 4 questions.

b) The remaining 20 marks are awarded through continuous evaluation of assignments / mini project in each subject as notified by the teacher at the beginning of the semester.

Students shall be informed regarding the comprehensive assignment/project during first week of semester and they have to submit completed assignment on or before 12^{th} week of semester.

10.1.2 Laboratory Courses: 25 marks

- For Laboratory courses there shall be continuous evaluation during the semester for 25 internal marks. The distribution of internal marks are given below:

Table 5: Distribution of Marks

Sl.No.	Criteria	Marks
1	Day to Day work	10
2	Record	05
3	Internal Examination	10

10.1.3 Seminar/Mini project: 25 marks

The distribution of internal marks for the seminar/mini project is given below.

Table 6: Distribution of Marks

Sl.No.	Criteria	Marks
1	Report	15
2	Seminar & Viva-viva	10

10.1.4 Major Project: (50 marks each in 3rd & 4th semesters)

The continuous internal evaluation for 50 marks allocated for the project work in each semester of 3rd & 4th shall be on the basis of two seminars by each student on the topic of his/her project evaluated by project review committee and day to day assessment by the supervisor in each semester. The project review committee consists of Head of Department, respective internal guide and three senior faculty members of the department. The distribution of marks is as follows.

Table 7: Continuous internal assessment in each semester

Sl.No.	Criteria	Marks
1	Two seminars	15+15
2	Day to day assessment	20

10.2 Semester End Examinations

10.2.1 Theory Courses: 60 marks

- The Semester end examinations shall be conducted for 3 hours duration at the end of the semester. The question paper shall be given in the following pattern :
- There shall be two questions from each unit with internal choice. Each question carries 15 marks. Each course shall consist of four units of syllabus.

10.2.2 Lab Courses: 50 marks

35 marks are allotted for experiments/job works and **10** marks are allotted for viva-voce examination and **5** marks for record.

10.2.3 Seminar/Mini project: 50 marks

There shall be a seminar presentation. For Seminar/Mini Project, a student under the supervision of a faculty member, shall collect the literature on a topic and critically review the literature and submit it to the Department in a report form and shall make an oral presentation before the Departmental Committee. The Departmental Committee consists of Head of the Department, supervisor and two other senior faculty members of the department. For Seminar/Mini Project the evaluation is done for 50 marks internally. A candidate has to secure a minimum of 50% to be declared successful.

10.3 Major Project:

The work on the project shall be initiated in the beginning of the second year and the duration of the project is two semesters. Every candidate shall be required to submit thesis or dissertation after taking up a topic approved by the Project Review Committee.

a. A Project Review Committee (PRC) shall be constituted with Head of the Department as chair person, two senior faculty members of the concerned department.

b. The candidate has to submit, in consultation with his project supervisor, the title, objective and plan of action of his project work to the Project Review Committee for its approval before the second semester end examinations. After obtaining the approval of the Committee the student can initiate the Project work after the second semester end examinations.

c. If a candidate wishes to change his supervisor or topic of the project he can do so with approval of the PRC. However, the Project Review Committee (PRC) shall examine whether the change of topic/supervisor leads to a major change of his initial plans of project proposal. If so, his date of registration for the project work starts from the date of change of Supervisor or topic as the case may be.

d. After approval of the topic in Project Review Committee, the candidate shall be required to submit status report in four stages. The first one in the mid of 3rd semester, second one in the end of 3rd semester, third one in the mid of 4th semester and the final report in the form of draft copy of thesis for the approval of PRC to the Head of the Department and shall make an oral presentation before the PRC.

e. Due weightage will be given to the papers published from the thesis submitted in the order of International Journal, National Journal, International conference and National conference while evaluating the thesis.

f. Three copies of the Project Thesis certified by the supervisor shall be submitted to the College.

g. The thesis shall be adjudicated by one external examiner selected by the Principal. For this, Head of the Department shall submit a panel of five examiners, who are eminent in the field.

h. The viva-voce examination shall be conducted by a board consisting of the supervisor, Head of the Department and the external examiner. Head of the Department shall coordinate and make arrangements for the conduct of viva-voce examination. If any candidate gets less than 50% marks in the viva-voce examination, he/she shall revise and resubmit the project work and reappear for viva-voce examination when next conducted.

In a special case, if any candidate does not submit his/her thesis due to ill health or any other reason permitted by the head of the institution, he/she will be given another chance to attend for the viva-voce examination conducted separately at a later date, if the expenditure for conducting the viva-voce is completely borne by the candidate.

11. CONDITIONS FOR PASS AND AWARD OF CREDITS FOR A COURSE

11.1 Conditions for Pass and award of Grades and Credits:

a) A candidate shall be declared to have passed in individual Theory/Drawing course if he/she secures a minimum of 50% aggregate marks (Internal & semester end examination marks put together), subject to a minimum of 40% marks in semester end examination.

b) A candidate shall be declared to have passed in individual lab/project course if he/she secures a minimum of 50% aggregate marks (Internal & semester end examination marks put together), subject to a minimum of 50% marks in semester end examination.

c) If a candidate secures minimum of 40% marks in Theory Courses in the semester end examination and 40% - 49% of the total marks in the semester end examination and internal evaluation taken together in some theory courses and secures an overall aggregate of 50% in all theory courses of that semester he/she declared to be passed in the theory courses of that semester.

d) The student has to pass the failed course by appearing the examination when offered next, as per the requirement for award of the degree.

e) On passing a course of a programme, the student shall earn assigned credits in that Course.

11.2 Method of Awarding Letter Grades and Grade Points for a Course.

A letter grade and grade points will be awarded to a student in each course based on his/her performance as per the grading system given below.

Table 8: Grading System for individual subjects/labs

Theory/Drawing	Lab/Project	Grade Points	Letter Grade
85-100%	85-100%	10	Ex
75-84%	75-84%	9	A+
70-74%	70-74%	8	A
65-69%	65-69%	7	B+
60-64%	60-64%	6	B
50-59%	55-59%	5	C
40-49%	50-54%	4	D
< 40%	< 50%	0	F (Fail)

11.3 Calculation of Semester Grade Points Average (SGPA)* and award of division for the program.

The performance of each student at the end of the each semester is indicated in terms of SGPA. The SGPA is calculated as below:

$$\text{SGPA} = \frac{\Sigma(CR \times GP)}{\Sigma CR} \quad \text{(for all courses passed in semester)}$$

Where CR= Credits of a course

GP = Grade points awarded for a course

*SGPA is calculated for the candidates who passed all the courses in that semester.

11.4 Calculation of Cumulative Grade Point Average (CGPA) for Entire Programme.

The CGPA is calculated as below:

$$\text{CGPA} = \frac{\Sigma(CR \times GP)}{\Sigma CR} \quad \text{(for entire programme)}$$

Where CR= Credits of a course

GP = Grade points awarded for a course

Table 9: Award of Divisions

CGPA	DIVISION
≥8	First Class with distinction
≥6 - <8	First Class
≥5 - <6	Second Class
≥4 - <5	Pass Class
<4	Fail

11.5 Transitory Regulations

A candidate, who is detained or discontinued in the semester, on readmission shall be required to pass all the courses in the curriculum prescribed for such batch of students in which the student joins subsequently. However, exemption will be given to those candidates who have already passed in such courses, in the earlier semester(s) as approved by Board of Studies and ratified by Academic Council.

11.6 Consolidated Grade Card

A consolidated grade card containing credits & grades obtained by the candidates will be issued after completion of the two year M.Tech Programme.

12. REVALUATION

- As per the notification issued by the Chief Controller of Examinations, the students can submit the applications for revaluation, along with the fee receipt for revaluation of his/her answer script(s) of theory course(s), if he/she is not satisfied with marks obtained.
- The Controller of Examinations shall arrange for revaluation of those answer script(s).
- A new external examiner, other than the first examiner, shall revaluate the answer script(s).
- Better marks of the two will be taken into consideration.

13. READMISSION CRITERIA

A candidate, who is detained in a semester due to lack of attendance/marks, has to obtain written permission from the Principal for readmission into the same semester after duly fulfilling all the required norms stipulated by the college in addition to paying an administrative fee of Rs. 1,000/-

14. BREAK IN STUDY

Student, who discontinues the studies for what so ever may be the reason, can get readmission into appropriate semester of M.Tech programme after break-in study only with the prior permission of the Principal of the College provided such candidate shall follow the transitory regulations applicable to such batch in which he/she joins. An administrative fee of Rs. 2000/- per each year of break in study in addition to the prescribed tuition and special fee has to be paid by the candidate to condone his/her break in study.

15. ELIGIBILITY FOR AWARD OF M.TECH. DEGREE

The M.Tech., Degree shall be conferred on a candidate who has satisfied the following requirement. A student should register himself for 80 Credits, and should obtain all the 80 credits in order to become eligible for the award of M.Tech Degree.

16. CONDUCT AND DISCIPLINE

- Students shall conduct themselves within and outside the premises of the Institute in a manner befitting the students of our Institution.
- As per the order of Honorable Supreme Court of India, ragging in any form is considered a criminal offence and is banned. Any form of ragging will be severely dealt with.
- The following acts of omission and/or commission shall constitute gross violation of the code of conduct and are liable to invoke disciplinary measures with regard to ragging.
    - Lack of courtesy and decorum; indecent behavior any where within or outside the campus.
    - Willful damage or distribution of alcoholic drinks or any kind of narcotics to fellow students /citizens.
- Possession, consumption or distribution of alcoholic drinks or any kind of narcotics or hallucinogenic drugs.
- Mutilation or unauthorized possession of library books.
- Noisy and unseemly behavior, disturbing studies of fellow students.
- Hacking computer systems (such as entering into other person's areas without prior permission, manipulation and/or damage of computer hardware and software or any other cyber crime etc.
- Students are not allowed to use cell phones in the campus.
- Plagiarism of any nature is prohibited.
- Any other act of gross indiscipline as decided by the college from time to time.
- Commensurate with the gravity of offense, the punishment may be reprimand, fine, expulsion from the institute / hostel, debarment from a examination, disallowing the use of certain facilities of the Institute, rustication for a specified period or even outright expulsion from the Institute, or even handing over the case to appropriate law enforcement authorities or the judiciary, as required by the circumstances.
- For an offence committed in (i) a hostel (ii) a department or in a class room and (iii) elsewhere, the Chief Warden, the Head of the Department and the Principal, respectively, shall have the authority to reprimand or impose fine.
- Cases of adoption of unfair means and/or any malpractice in an examination shall be reported to the Principal for taking appropriate action.
- Un authorized collection of money in any form is strictly prohibited.
- Detained and Break-in-Study candidates are allowed into the campus for academic purposes only with permission from Authorities.

- Misconduct committed by a student outside the college campus but having the effect of damaging, undermining & tarnishing the image & reputation of the institution will make the student concerned liable for disciplinary action commensurate with the nature & gravity of such misconduct.
- The Disciplinary Action Committee constituted by the Principal, shall be the authority to investigate the details of the offence, and recommend disciplinary action based on the nature and extent of the offence committed.
- "Grievance appeal Committee" (General) constituted by the Principal shall deal with all grievances pertaining to the academic / administrative /disciplinary matters.
- All the students must abide by the code and conduct rules of the college.

17. MALPRACTICES

- The Principal shall refer the cases of malpractices in internal assessment tests and Semester-End Examinations, to a Malpractice Enquiry Committee, constituted by him/her for the purpose. Such committee shall follow the approved scales of punishment. The Principal shall take necessary action, against the erring students basing on the recommendations of the committee.

- Any action on the part of candidate at an examination trying to get undue advantage in the performance or trying to help another, or derive the same through unfair means is punishable according to the provisions contained hereunder. The involvement of the Staff, who are in charge of conducting examinations, valuing examination papers and preparing/keeping records of documents relating to the examinations in such acts (inclusive of providing incorrect or misleading information) that infringe upon the course of natural justice to one and all concerned at the examination shall be viewed seriously and recommended for award of appropriate punishment after thorough enquiry.

18. OTHER MATTERS

- The physically challenged candidates who have availed additional examination time during their B.Tech/PGCET examinations will be given additional examination time on production of relevant proof/documents.

- Students who are suffering from contagious diseases are not allowed to appear either internal or semester end examinations.

- The students who participated in coaching/tournaments held at state/National /International levels through University / Indian Olympic Association during end semester external examination period will be promoted to subsequent semesters till the entire course is completed as per the guidelines of University Grants Commission Letter No. F.1-5/88 (SPE/PES), dated 18-08-1994.

- The Principal shall deal with any academic problem, which is not covered under these rules and regulations, in consultation with the Heads of the Departments in an appropriate manner, and subsequently such actions shall be placed before the academic council for ratification. Any emergency modification of regulation, approved in the Heads of the Departments Meetings, shall be reported to the academic council for ratification.

19. AMENDMENTS TO REGULATIONS

The Academic Council may, from time to time, revise, amend or change the regulations, schemes of examination and/or syllabi.

20. SCHEME OF INSTRUCTION

M.Tech Computer Science & Engineering Course Structure VR10

First Year Semester I

S.No	Sub. Code	Subject Title	L	T	P	C	I	E	T
1	CSCS1001	Mathematical Foundation for Computer Science	4	-	--	4	40	60	100
2	CSCS1002	Advanced Data Structures & Algorithms	4	-	--	4	40	60	100
3	CSCS1003	Computer Organization & Architecture	4	--	--	4	40	60	100
4	CSCS1004	Web Technologies	4	--	--	4	40	60	100
5	CSCS1005	Advanced Operating Systems	4	--	--	4	40	60	100
6	CSCS1006	Elective – I A - Advanced DBMS B - Digital Image Processing C – AI and Neural Networks D – Parallel Computing & Alg.	4	--	--	4	40	60	100
7	CSCS1051	Advanced Data Structures Lab	--	--	3	2	25	50	75
8	CSCS1052	Web Technologies Lab	--	--	3	2	25	50	75
			24	**--**	**6**	**28**	**290**	**460**	**750**

First Year Semester II

S.No	Sub. Code	Subject Title	L	T	P	C	I	E	T
1	CSCS2001	Object Oriented Analysis and Design	4	-	--	4	40	60	100
2	CSCS2002	Advanced Computer Networks	4	-	--	4	40	60	100
3	CSCS2003	Embedded Computing Systems	4	--	--	4	40	60	100
4	CSCS2004	Data Mining	4	--	--	4	40	60	100
5	CSCS2005	Elective – II A – S/W Testing Methodologies B – Pattern Recognition C – Natural Language Processing D – Human Computer Interaction	4	--	--	4	40	60	100
6	CSCS2006	Elective – III A – Cryptography and Network Security B – Secured Database Application Development. C – Bio-informatics D – Soft Computing	4	--	--	4	40	60	100
7	CSCS2051	Software Engineering Lab	--	--	3	2	25	50	75
8	CSCS2052	Data Mining Lab	--	--	3	2	25	50	75
			24	**--**	**6**	**28**	**290**	**460**	**750**

L: Lecture **T**: Tutorial **P**: Practical **C**: Credits **I**: Internal Assessment **E**: End Semester **T**: Total Marks

Second Year Semester III and IV

S.No	Sub. Code	Subject Title	L	T	P	C	I	E	T
1	CSCS4051	Project Work				24	100	200	300
						24	**100**	**200**	**300**

21. CATEGORIES OF COURSES AND THEIR DISTRIBUTION

S.No.	Courses	Category	Credits
4.	Programme Core Courses	PC	40
5.	Elective Courses	EL	12
6.	Department/Programme Major Project	MP	24
7.	Supporting to the Core	SC	4
	Total Credits		80

Courses Distribution: Category wise

Year & Semester	Programme Core	Elective	Major Project	Supporting to the Core	Total Credits
I Year I Semester	CSCS1002-4 CSCS1003-4 CSCS1004-4 CSCS1005-4 CSCS1051-2 CSCS1052-2	CSCS1006-4		CSCS1001-4	**28**
I Year II Semester	CSCS2001-4 CSCS2001-4 CSCS2003-4 CSCS2004-4 CSCS2051-2 CSCS2052-2	CSCS2005-4 CSCS2006-4			**28**
II Year III & IV Semesters			CSCS4051-24		**24**
Total Credits:					**80**

22. DETAILED SYLLABUS:

CSCS1001
MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- Builds Foundations for Computer Science and Computer Engineering.
- Develops how to symbolize, read and understand the logical arguments also construct the logical arguments.
- Analyzes relations – binary, partial order. Construct Hasse diagrams.
- Introduces counting techniques.
- Introduces fundamentals in graph theory.
- Understand the essence of computing through simple models of computational devices
- Understand the limitations of computing, the relative power of formal languages

Learning Outcomes:

- Symbolize the Statements, Validate the arguments, verify the proofs and construct proofs
- Understand relations, analyzes binary relations, reflexive, symmetric, transitive and partial ordered relations.
- Solve different types of recurrence relations and applies wherever necessary.
- Gain knowledge about graphs and graph coloring
- Appreciate the role of abstract models in theory building
- Have better skills to communicate formal arguments and proofs

UNIT I:

Foundations: Sets, Relations & functions, Proof & Problem Solving, Fundamentals of Logic, Logical Inferences, Methods of Proof of an implication, First Order Logic & Other methods of proof, Rules of Inference and quantified propositions, Mathematical Induction.

UNIT II:

Recurrence Relations: Generating Functions, Solving recurrence relations, the methods of characteristics roots, undetermined coefficient method.

Relations & Digraphs: Equivalence relations, ordering relations, Lattices & enumerations, Operations on Relations, Paths & Closures, Directed graphs & Adjacency matrices.

UNIT III:

Automata: Introduction to Automata, The central concepts of automata theory - Alphabets, Strings, Languages.

Finite Automata: An Informal picture of finite automata, Deterministic finite automata (DFA) - Definition of DFA, DFA processing strings, Notations for DFA, Extended transition function, the language of DFA, Non deterministic finite automata (NFA) – Definition of NFA, Extended transition function, the language of NFA, Equivalence of DFA and NFA Finite Automata with $\in$ transitions: Use of $\in$ - transition, notation for an $\in$ - NFA, Epsilon closures, extended transitions and languages, Applications

UNIT □ IV

Regular Expressions and Languages: Regular expressions, finite automata and regular expressions, Algebraic laws of regular expressions, Proving languages are not regular – Pumping lemma for regular languages, Applications of the pumping lemma, Closure Properties of Regular Languages, Equivalence and minimization of automata – Minimization of DFA.

Context Free Grammars: Context Free Grammars, Parse Trees, Constructing parse trees, derivations

and parse trees, ambiguous grammars. Context –Free languages - Normal form's for context- Free grammars, the pumping lemma for context free languages.
Pushdown Automata: Definition of the Pushdown automata, the languages of PDA, Equivalences of PDA's and CFG's.

Learning Resources:

Text Books:

1. Joe L. Mott, Abrabam Kandel and Theodore P. Bakev, *Discrete Mathematics for Computer Scientists and Mathematics*. PHI. (UNIT I & II)
2. John.E. Hopcroft, R. Motwani, and Jeffery. D. Ullman, *Introduction to Automata Theory, Languages and Computations*. 2 ed, Pearson Education, 2003 (UNIT III & IV)

Reference Books:

1. C. L. Liu, *Elements of Discrete Mathematics*.
2. Rosen, *Discrete Mathematics*
3. K. L. P. Mishra and N. Chandrasekharan *Theory of Computation*. PHI.
4. Ralph. P. Grimaldi, *Discrete and Combinational Mathematics- An Applied Introduction*. 5 ed, Pearson Education.
5. Trembly J.P. and Manohar .P, *Discrete Mathematical Structures with applications to Computer Science*. TMH.

CSCS1002
ADVANCED DATA STRUCTURES AND ALGORITHMS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- Analyze step by step and develop algorithms to solve real world problems.
- Implementing various data structures
- Investigate whether the algorithm found is the most efficient.
- Formulate the time order analysis for an algorithm.
- Formulate the space needs for the implementation of an algorithm.
- Prove the correctness of an algorithm

Learning Outcomes:

Upon completion of this course the student will be able to:

- Differentiate static and dynamic data structures and their implementation
- Implement appropriate data structure for a given application
- Know the basic ingredients of a greedy algorithm, and how to approach arguing the correctness of such algorithms
- Familiar with dynamic-programming algorithms, how to apply them via both memorization and tables, and recognize when a dynamic programming approach might yield a good solution to a problem.
- Familiar with Backtracking, Branch and Bound technique, solution of n queen problems and traveling sales problem.

UNIT I:

Overview of Data Structures: Review of Arrays, Stacks, Queues, linked lists, Linked stacks and Linked queues, Applications.

Algorithm Analysis: Efficiency of algorithms, Apriori Analysis, Asymptotic Notations, Time complexity of an algorithm using O notation, Polynomial Vs Exponential Algorithms, Average, Best, and Worst Case Complexities, Analyzing Recursive Programs.

Trees: Introduction, Definition and Basic terminologies of trees and binary trees, Representation of trees and Binary trees, Binary tree Traversals.

Graphs: Introduction, Definitions and Basic technologies, representation of Graphs, Graph Traversals and applications.

UNIT II:

Binary Search Trees, AVL Trees and B Trees:

Binary Search Trees: Definition, Operations and applications. AVL Trees: Definition, Operations and applications. B-Trees: Definition, Operations and applications.

Red Black Trees, Splay Trees: Operations and applications.

Hash Tables: Introduction, Hash Tables, Hash Functions and applications

UNIT III:

Divide and Conquer & Greedy Method: Divide – and – Conquer: General Method, Binary Search, Finding Maximum and Minimum, Quick Sort, Merge sort, Strassen's Matrix Multiplication, Greedy Method: General Method, Minimum Cost Spanning Trees, Single Source Shortest Path, Knapsack. Dynamic Programming General Method, All Pairs Shortest Path, Single Source Shortest Path, 0 / 1 Knapsack problem, Optimal Binary Search Trees.

UNIT IV

Back Tracking and Branch and Bound: General Method, 8 – Queen's Problem, Graph

Coloring. Branch – and – Bound: The Method, LC Search, Control Abstraction, Bounding, 0 / 1 Knapsack Problem.

NP Hard and NP Complete: Cook's theorem

Learning Resources:

Text Books:

1. G. A. V. Pai, *Data Structures and Algorithms*. TMH, 2009.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, *Fundamentals of Computer Algorithms*. 2 ed, University Press.

Reference Books:

1. Thomas Cormen, C. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*. 2 ed, PHI.
2. Aho, Hopcraft and Ullman, *Design and Analysis of Computer Algorithm*. PEA, 1998.
3. E. Horowitz and S. Sahani, *Design and Analysis of Algorithms*. 3 ed, Galgotia.

CSCS1003
COMPUTER ORGANIZATION AND ARCHITECTURE

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: **Students learn:**

- Basic building blocks of the computer.
- Functional units: operation, interaction and communication.
- The factors and trade-offs that affect computer performance.
- Concrete representation of data at the machine level.
- Machine level Computations

Learning Outcomes: **Upon completion of this course the student will be able to:**

- Understand the fundamental architectural principles.
- Measure the performance of modern microprocessor designs.
- Construct alternative computer architecture designs.
- Acquire knowledge about CPU, memory and storage systems.

UNIT I:

Computer Structure – Hardware, software, system software, Von-neumann architecture – case study. IA -32 Pentium: registers and addressing, instructions, assembly language, program flow control, logic and shift/rotate instructions, multiply, divide MMX, SIMD instructions, I/O operations, subroutines.

Input/Output Organization - interrupts, DMA, Buses, Interface circuits, I/O interfaces, device drivers in windows, interrupt handlers

UNIT II:

Processing Unit - Execution of a complete instruction, multiple bus organization, hardwired control, micro programmed control.

Pipelining: Data hazards, instruction hazards, influence on instruction sets, data path & control consideration, RISC architecture introduction.

UNIT III:

Memory: Types and hierarchy, model level organization, cache memory, performance considerations, mapping, virtual memory, swapping, paging, segmentation, replacement policies.

Processes and Threads: Processes, threads, inter process communication, classical IPC problems, Deadlocks.

UNIT IV

File system: Files, directories, Implementation, UNIX file system

Security: Threats, intruders, accident data loss, basics of cryptography, user authentication.

Learning Resources:

Text Books:

1. Carl Hamacher, Z. Vranesic and Safea Zaky, *Computer Organization*. 5 ed, McGraw Hill.
2. Andrew S. Tanenbaum, *Modern Operating Systems*. 2 ed, Pearson/PHI.

Reference Books:

1. William Stallings, *Computer Organization and Architecture*. 6 ed, Pearson/PHI.

2. Morris Mano, *Computer System Architecture*. 3 ed, Pearson Education.
3. Abraham Silberchatz, Peter B. Galvin and Greg Gagne, *Operating System Principles*. 7 ed , John Wiley
4. Stallings, *Operating Systems –Internals and Design Principles*. 5 ed, Pearson Education/PHI, 2005

CSCS1004
WEB TECHNOLOGIES

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: **Students Learn**

- Web site planning, management and maintenance.
- Developing the web pages with the help of frames, scripting languages, and evolving technology like DHTML, XML.
- Developing applications using Java GUI components
- The concept of designing secure web pages using java based technologies like Servlets and JSP.
- Developing web sites which are secure and dynamic in nature and writing scripts which get executed on server as well.

Learning Outcomes: At the end of this course the students will be able to

- Design web pages using standard web deigning tools like HTML, DHTML and server based technologies like Servlets and JSP.
- Design Java applications using GUI components such as Applets and Swings.
- Understand how java GUI components may access enterprise data bases.
- Understand the use of APIs in robust, enterprise three level application developments.
- Understand the Java features for secure communications over the internet

UNIT I:
HTML Common tags - List, Tables, images, forms, Frames, Cascading Style sheets, Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script.
XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX
UNIT II:
Review of Applets: Event Handling, AWT Programming.
Introduction to Swing: Handling Swing Controls like Icons – Labels – Buttons – Text Boxes – Combo – Boxes – Tabbed Panes – Scroll Panes – Trees – Tables Differences between AWT Controls & Swing Controls, JApplet, Developing a Home page using Applet & Swing.
UNIT III:
Web servers: Tomcat Server installation & Testing.
Servlets & Application Development: Lifecycle of a Servlet, JSDK, The Servlet API, The javax.servlet Package, Reading Servlet parameters, Reading initialization parameters, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues.
Database Access: Database Access, Database Programming using JDBC Studying Java.sql package, accessing a database from a servlet application.

UNIT □ IV
Introduction to JSP: The Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC architecture. AJAX.
JSP Application Development: Generating Dynamic Content, Using Scripting Elements, Implicit JSP Objects, Conditional Processing, Declaring Variables and Methods, Sharing Data between JSP pages, Sharing Session and Application Data – Memory Usage Considerations, Accessing a Database from a JSP Page Application.

Learning Resources:

Text Books:

1. Dietel & Dietel, *Internet & World Wide Web.* PHI.
2. Patrick Naughton and Herbert Schildt, *The complete Reference Java 2.* 5 ed, TMH (Chapters: 25) (UNIT 2,3)
3. Hans Bergsten and SPD O'Reilly, *Java Server Pages.* (UNITs 3,4,5)

Reference Books:

1. Marty Hall and Larry Brown, *Programming world wide web- Serdets and Java Server Pages Volume 1: Core Technologies.* Pearson

CSCS1005
ADVANCED OPERATING SYSTEMS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- Expose students to current and classical operating systems literature
- To get a comprehensive knowledge of the architecture of distributed systems.
- To get a knowledge of multiprocessor operating system and database operating systems, Real time systems
- Give students an understanding of what it means to do research in computer science and specifically operating systems
- Teach students to critically evaluate research papers

Learning Outcomes:

- Students will be familiar with advanced operating systems literature
- Design issues of advanced operating systems and mechanisms to build these systems
- Learns the differences between general purpose and database operating systems
- Learns how to tackle some challenging projects and write paper suitable for publication

UNIT I:

Overview: Introduction, functions of operating systems, design approaches, why advanced operating systems, types of advanced operating systems.

Architectures of Distributed Systems - System Architecture types, distributed operating systems, issues in distributed operating systems

Theoretical Foundations - inherent limitations of a distributed system – lamp ports logical clocks – vector clocks – casual ordering of messages – global state – cuts of a distributed computation – termination detection.

Distributed resource management: introduction-architecture – mechanism for building distributed file systems – design issues

UNIT II:

Distributed shared memory-Architecture–algorithms for implementing DSM–memory coherence and protocols–design issues.

Distributed Scheduling – introduction – issues in load distributing – components of a load distributing algorithm –load distributing algorithm – selecting a suitable load sharing algorithm – requirements for load distributing.

Failure Recovery: introduction– basic concepts – classification of failures – backward and forward error recovery, backward error recovery- recovery in concurrent systems – consistent set of check points – check pointing for distributed database systems- recovery in replicated distributed databases

UNIT III:

Multiprocessor System Architectures- basic multiprocessor system architectures – inter connection networks for multiprocessor systems – caching – hypercube architecture.

Multiprocessor Operating System - structures of multiprocessor operating system, operating system design issues- threads- process synchronization and scheduling.

Database Operating systems: Introduction- What is different-requirements of a database operating system.

UNIT IV

Real time systems: Typical real time applications: Digital Control, High level Controls, signal processing, other real time applications

Hard Vs Soft real time systems: Jobs and processors, Release times, deadlines, and timing constraints, Hard and soft timing constraints, Hard real time systems, Soft real time systems

A reference model of real time systems: Processors and resources, Temporal parameters of real time workload, periodic task model, precedence constraints and data dependency, Scheduling hierarchy

Approaches to real time scheduling: Clock driven approach, weighted round robin approach, Priority Driven approach, Dynamic Vs Static systems, effective release times and deadlines

Learning Resources:

Text Books:

1. Mukesh Singhal and Niranjan G. Shivaratri, *Advanced concepts in operating systems: Distributed, Database and Multiprocessor Operating Systems.*, TMH, 2001**(Unit I, II, III)**
2. Jane W.S.Liu, *Real Time Systems*. Pearson Education Asia **(Unit IV)**

Reference Books:

1. AndrewS.Tanenbaum, *Modern Operating System*. PHI, 2003.
2. Pradeep K.Sinha, *Distributed Operating System-Concepts and Design*, PHI, 2003.
3. C. M. Krishna and G. Shin, *Real Time Systems*. Tata McGraw Hill 1997.

CSCS1006 A
ADVANCED DATABASE MANAGEMENT SYSTEMS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: Students Learn

- Distributed Database Processing
- Distributed Database Systems and Design
- Parallel Database Systems
- Distributed Object Database Management Systems

Learning Outcomes: Upon completion of this course the students will be familiar with:

- Distributed Database Processing.
- Query Optimization Techniques
- In depth knowledge of Distributed Database Systems and Distributed Database Design.
- Query Optimization Techniques
- Parallel Database Systems
- Distributed Object Database Management Systems
- Scope of Research in Query Optimization in various Architectural Models

UNIT I:

Introduction: Distributed Data Processing, Distributed Database System, Promises of DDBS, Complicating Factors, Problem Areas.

Overview of Relational DBMS: Relational Databases Concepts, Normalization, Integrity rules, Relational data languages.

Distributed DBMS Architecture: DBMS Standardization, Architectural Models for Distributed DBMS, DDBMS Architecture.

Distributed Database Design: Alternative Design Strategies, Distribution Design Issues, Fragmentation, and Allocation.

UNIT II:

Overview of Query Processing: Objectives of Query Processing, Characterization of Query Processors, Layers of Query Processing.

Query Decomposition and Data Localization: Query Decomposition, Localization of Distributed Data

Optimization of Distributed Queries: Query Optimization, Centralized Query Optimization, Distributed Query Optimization Algorithms.

UNIT III:

Transaction Management: Definition of a Transaction, Properties of transaction, Types of transactions.

Distributed Concurrency Control:. Serialization, Concurrency Control Mechanism and Algorithms. Time stamped and Optimistic concurrency control Algorithms, Dead lock Management.

Distributed DBMS Reliability: Reliability concepts and Measures, Fault-tolerance in Distributed systems, failures in Distributed DBMS, Local & Distributed Reliability Protocols.

UNIT IV

Parallel Database Systems: Database Servers, Parallel Architecture, Parallel DBMS Techniques, Parallel Execution Problems, Parallel Execution for Hierarchical architecture

Distributed Object Database Management Systems: Fundamental object concepts and Models, Object Distributed Design, Architectural Issues, Object Management, Distributed Object Storage, Object Query Processing.

Learning Resources:

Text Book:

1. M. Tamer Ozsu and P. Valduriez, *Principles of Distributed Database Systems*. 2 ed, Pearson, 2001.

Reference Books:

1. Stefano Ceri Giuseppe Pelagatti, *Principles of Distributed Database Systems*. TMH.
2. Raghuramakrishnan and Johhanes Gehrke, *Database Management Systems*. 3 ed, TMH.

CSCS1006B
DIGITAL IMAGE PROCESSING

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- To introduce students to the Basic concepts and analytical methods of analysis of digital images.
- To Study fundamental concepts of Digital Image Processing and basic relations among pixels.
- To Study different Spatial and Frequency domain concepts.
- To understand Restoration process of degraded image and Multi resolution processing.
- To understand image compression and Segmentation Techniques.

Learning Outcomes: Up on completion of this course the students can be able to understand

- Basic concepts and analytical methods of analysis of digital images.
- Fundamental concepts of Digital Image Processing and basic relations among pixels.
- Different Spatial and Frequency domain concepts.
- Restoration process of degraded image and Multi resolution processing.
- Image compression and Segmentation Techniques.

UNIT I:

Introduction: Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System.

Digital Image Fundamentals: Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Some basic Relationships between Pixels.

Image Enhancement in the Spatial Domain: Some Basic Gray Level Transformation, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing spatial Filters, Sharpening spatial Filters.

UNIT II:

Image Enhancement in the Frequency Domain: Introduction to the Fourier Transform and the Frequency Domain, Smoothing frequency-domain Filters, Sharpening Frequency-domain Filters, Homomorphic Filtering, Implementation.

Image restoration: A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function, Color Image Processing Introduction.

UNIT III:

Image Segmentation: Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation,

Image Compression: Fundamentals, image compression models, error-free compression, lossy predictive coding, image compression standards

UNIT IV

Wavelets and Multiresolution Processing: Multiresolution Expansions, Wavelet Transforms in one Dimension, The Fast Wavelet Transform, Wavelet Transforms in Two Dimensions

Morphological Image Processing: Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms, Object Recognition: Patterns and patterns classes.

Learning Resources:

Text Books:

1. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*. 2 ed, PHI/Pearson Education.

Reference Books:

1. A.K.Jain, *Fundamentals of Digital Image Processing*. Prentice Hall India.
2. Madhuri. A. Joshi, *Digital Image Processing*. PHI.
3. Milan Sonka, Vaclav Hlavac and Roger Boyle, *Image Processing, Analysis and Machine Vision*. 2 ed, Thomson Learning.

CSCS1006C
ARTIFICIAL INTELLIGENCE AND NEURAL NETWORKS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: Students Learn

- Definition and origins of AI.
- Basic knowledge of neural networks and expert systems
- Wide variety of neural network models and their applications.
- Creating and using neural network models.
- Some mathematical understanding of neural network models.
- Computational tools to perform experiments leading to new theoretical insights.

Learning Outcomes: By the end of this course, students will

- Understand the principles of problem solving and be able to apply them successfully.
- Be familiar with techniques for computer-based representation and manipulation of complex information, knowledge, and uncertainty.
- Gain awareness of several advanced AI applications and topics such as intelligent agents, planning and scheduling, machine learning.
- Understand various types of neural networks used for purposes such as in discriminators, classifiers, computation.
- Understand how neural networks are implemented using training algorithms such as feedforward, back-propagation.

UNIT I:

Problems, Problem Spaces And Search: Defining the Problem as a State space Search, Production Systems, Problem Characteristics, Production system characteristics, Issues in the Design of Search Programs.

Heuristic Search Techniques: Generate-and-test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis

UNIT II:

Using Predicate Logic: Representing Simple Facts in logic, Representing Instance and Isa Relationships, Computable Functions and Predicates, Resolution, Natural Deduction.

Weak Slot-and-Fillers Structures: Semantic Nets, Frames

Strong Slot-and- Fillers Structures: Conceptual Dependency, Scripts

UNIT III:

Overview – applications - Engg approaches in NN - Biological Inspiration - Learning Approaches - Mathematical Fundamentals for ANN Study.

Elementary building blocks - biological neural units - artificial unit structures - unit net activation to output characteristics - artificial unit model extensions - single unit mappings and the perceptron - linear seperability – perceptrons - adaline – madaline - MLP

NN based pattern associators - PA design and evaluation - linear associative mappings.

UNIT IV

Recurrent Networks - Hopfield Network design - applications to optimization problem - Application to Traveling salesman problem – BAM.
Self Organizing Networks – Introduction – clustering – kohonen – ART networks – Learning Theroies – Dynamics – Applications.
RBF – Design – Training – Applications – TDNN – Learning – Applications.

Learning Resources:

Text Books:

1. Elaine Rich and Kevin Knight, *Artificial Intelligence.* 2 ed, TMH.
2. Robert J Schalkoff, *Artificial Neural Networks.* TMH, 1997.

Reference Books:

1. Patrick Henry Winston, *Artificial Intelligence.*, Pearson Education/Prentice Hall of India.
2. Russel and Norvig, *Artificial Intelligence.* Prentice Hall of India/Pearson Education.
3. Giarratano, *Expert Systems :Principles and Programming.* Cengage Publications
4. B.Vegnanarayana, *Artificial Neural Networks.* PHI, 2005.
5. Li Min Fu, *Neural Networks in Computer Intelligence.* TMH, 2003.
6. Jacek M.Zurada, *Artificial Neural Networks Systems.* Jaico, 2006.

CSCS1006D
PARALLEL COMPUTING AND ALGORITHMS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- To develop structural intuition of how the hardware and the software work, starting from simple systems to complex shared resource architectures
- To familiarize with the main parallel programming techniques.
- Study basic concepts of the discipline, e.g., of program, process, thread, concurrent execution, parallel execution and granularity
- Parallel algorithms for sorting and searching

Learning Outcomes:

- Understand fundamental parallel algorithms and how to incorporate high level parallelism into hardware and software.
- Designing, analyzing and implementing parallel algorithms suitable for execution on real parallel computers.
- Understand how to make efficient use of emerging parallel computer technology

UNIT I:

Introduction: Computational demands of Modern Science, advent of parallel processing, terminology-pipelining, data parallelism and control parallelism-Amdahl's. Basic parallel random access Machine Algorithms-definitions of P, NP and NP-Hard, NP-complete classes of sequential algorithms-NC –class for parallel algorithms.

Parallel Computer Architectures: Processor Arrays, Multi processors and multi computers. Mapping and scheduling aspects of algorithms- Mapping into meshes and hyper cubes, Load balancing, Graham's List scheduling algorithm, Coffman-graham scheduling algorithm for parallel processors.

UNIT II:

Elementary Parallel algorithms: Parallel algorithms on SIMD and MIMD machines, Analysis of these algorithms. Matrix Multiplication algorithms on SIMD and MIMD models.

Fast Fourier Transform algorithms- Implementation on Hyper cube Multi computer. Solving linear file - system of equations, parallelizing aspects of sequential methods back substitution and Tri diagonal.

UNIT III:

Parallel sorting methods: Odd-even transposition sorting on processor arrays. Biotonic-merge sort on shuffle –exchange ID, Array processor, 2D-Mesh processor and Hypercube Processor Array. Parallel Quick-sort on Multi processors. Hyper Quick sort on hypercube multi computers.

Parallel Searching: Ellis algorithm and Manber and ladner's Algorithms for dictionary operations.

UNIT IV

Parallel algorithms for Graph searching: All Pairs shortest paths and minimum cost spanning tree. Parallelization aspects of combinatorial search algorithms with Focus on Branch and Bound Methods and Alpha-beta Search methods.

Learning Resources:

Text Books:

1. Michael J. Quinn, *Parallel Computing Theory and Practice*, 2 ed, TMH.

REFERENCE BOOKS:

1. Guy E. Blelloch, *Programming Parallel Algorithms*. Communications of the ACM
2. Michael T Heath, Abhiram Ranade and Schreiber, *Algorithms for Parallel Processing*. Springer.
3. Seyed H Roosta, *Parallel Processing and Parallel Algorithms: Theory and Computation*. Springer.

CSCS1051
ADVANCED DATA STRUCTURES LAB

Lecture :	-	**Internal Assessment:**	25
Tutorial :	-	**Final Examination:**	50
Practical :	3 hrs/ Week	**Credits:**	2

Objectives:

- This course presents fundamental and advanced techniques to data structures.
- By the end of the course, the student should learn how to uses C programming language to implement a variety of data structures.

Learning Outcomes:

- Develop different linear and non-linear data structure programs
- Implementing the applications of linear and non linear data structures.
- Implementing the various sorting techniques using divide and conquer approach.
- Develop various graph algorithms using dynamic programming techniques
- Implementing the n-queens problem using backtracking techniques.

List of Laboratory Experiments:

Week1:

Write a C program to implement the Stacks and Queues Using Linked List.

Write a C program to implement all Stack Applications.

Week 2:

Write a C program to implement the Graph traversal techniques:

a) BFS b) DFS

Write a C program to implement the Binary Searching Tree ADT:

a) Insertion b) Deletion c) Traversal Techniques

Week 3 :

Write a C program to implement AVL tree using ADT.

a) Insertion b) Deletion

Week 4 :

Write a C program to implement the Red-Block Tree ADT:

a) Insertion b) Deletion c) Print

Week 5:

Write a C Program to implement the Hashing Techniques:

a) Open addressing b) Separate Chaining

Write a C Program to implement the sorting techniques using divide and conquer approach.

a) Merge sort b) Quick Sort

Week 6:

Write a C program to implement the Minimal Spanning Trees algorithms:

a) Kruskal b) Prims

Week 7:

Write a C program to implement the Single –Source Shortest Paths algorithms :

a) Dijkstra's b) Bellman-Ford

Write a C program to implement the Topological sorting Technique.

Week 8:

Write a C program to implement the following Dynamic programming techniques.

a)All-Pairs-Shortest-Path

Week 9:

Write a C program to implement the Optimal Binary Search Tree using Dynamic Programming Technique.

Week 10:

Write a C program to implement four queens problem and graph coloring using Back Tracking Technique.

Learning Resources:

Text Books:

1. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C. 2 ed,* Pearson Education.

CSCS1052
WEB TECHNOLOGIES LAB

Lecture :	-	**Internal Assessment:**	25
Tutorial :	-	**Final Examination:**	50
Practical :	3 hrs/ Week	**Credits:**	2

Objectives: **After studying the course the student can be able to:**

- Design web pages with the help of frames, scripting languages, and evolving technology like DHTML, XML.
- Develop applications using Java GUI components
- Understand the concept of designing secure web pages using java based technologies like Servlets and JSP.

Learning Outcomes: At the end of this course the students are able to

- Design web pages using standard web deigning tools like HTML, DHTML and server based technologies like Servlets and JSP.
- Design Java applications using GUI components such as Applets and Swings.
- Understand how java GUI components may access enterprise data bases.
- Understand the use of APIs in robust, enterprise three level application developments.
- Understand the Java features for secure communications over the internet

List of Programs:

Week # 1:
Design a web page that reflects the details of a College using simple HTML tags.

Week # 2:
Design a web application using different types of CSS.

Week # 3:
Design a student database using XML and display the content using XSL by validating through XML schema.

Week # 4:
Design a user registration form using Java Applets.

Week # 5:
Develop a menu based application page using java swings.

Week # 6:
Develop a home page for college using Swings.

Week # 7:
Design a simple java servlet application to retrieve the data from a client form and display the data.

Week # 8:
Design a java application to perform all the DML operations on a database.

Week # 9:
Design a web application for user management using Java servlets.

Week # 10:

Design a simple JSP application to perform simple operations.

Week # 11:

Design a web application for user management using JSP.

Week # 12:

Design a web application to share the data between multiple pages using sessions and cookies.

Learning Resources:

Text Books:

1. Dietel and Dietel, *Internet & World Wide Web.* PHI.
2. Patrick Naughton and Herbert Schildt., *The complete Reference Java 2. 5 ed*, TMH.
3. Hans Bergsten, SPD O'Reilly, *Java Server Pages.*

Reference Books:

1. Marty Hall and Larry Brown, *Programming World Wide Web-Sebesta, Pearson Core Servlets Andjavaserver Pages Volume 1: Core Technologies.* Pearson.

CSCS2001
OBJECT ORIENTED ANALYSIS AND DESIGN

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: This course introduces the students to Learn:

- About unified process.
- How software development is represented with the help of UML
- UML Diagram notations.
- Utilization of UML diagrams in the Software Development.

Learning Outcomes: Upon completion of the syllabus the student will be able to:

- Understands the benefits of Object Oriented Software Engineering
- Design their own projects with aids of UML Diagrams
- Know how to use inheritance in an effective way – in particular – in how they are espoused in design patterns
- Identify classes in their problem domain with a technique much better than finding nouns and verbs.
- Understand which code qualities are essential for writing maintainable code

UNIT I:
What is UML? The birth of UML; What is the Unified Process? The birth of UP; The requirements workflow. Software requirements – Metamodel; Use case modeling; UP activity: Find actors and use cases; Advanced use case modeling; The analysis workflow; Objects and classes

UNIT II:
Finding analysis classes; Relationships; Inheritance and polymorphism Generalization; Analysis packages; Use case realization; Advanced use case realization

UNIT III:
Activity diagrams; Advanced activity diagrams; The design workflow; Design classes; Refining analysis relationships; Interfaces and components

UNIT IV
Use case realization-design; State machines; Advanced state machines; The implementation workflow; Deployment

Learning Resources:

Text Books:

1. Jim Arlow, Ila Neustadt, *UML 2 and the Unified Process Practical Object-Oriented Analysis and Design*. 2 ed, Pearson Education, 2005.

Reference Books:

1. Booch, Rumbaugh and Jacobson, *The Unified Modeling Language Users Manual*, Pearson Education, 1999.
2. James Rumbaugh, Jacobson and Booch, *Unified Modeling Language Reference Manual*. Pearson Education.
3. Jacobson et al., *The Unified Software Development Process*, Pearson Education, 1999.
4. Simon Bennett, Steve McRobb and Ray Farmer *Object-Oriented Systems Analysis and Design Using UML*. 2 ed, TMH.

CSCS2002
ADVANCED COMPUTER NETWORKS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- This Course aims to provide a broad coverage of some new advanced topics in the field of computer networks.
- To study the problematic of service integration in TCP/IP networks focusing on protocol design, implementation and performance issues.
- To debate the current trends and leading research in the computer networking area.

Learning Outcomes:

- Understand the main abstract concepts related to the layered communication architecture.
- Understand basics and principles of new generation of computer networks
- To master the concepts of protocols, network interfaces, and design/performance issues in local area networks, wide area networks, Wireless networks and mobile networks.

UNIT I:

Review of Computer Networks and the Internet: What is the Internet, The Network edge, The Network core, Access Networks and Physical media, ISPs and Internet Backbones, Delay and Loss in Packet-Switched Networks, History of Computer Networking and the Internet - **Networking Devices:** Multiplexers, Modems and Internet Access Devices, Switching and Routing Devices, Router Structure. **The Link Layer and Local Area Networks:** Link Layer: Introduction and Services, Multiple Access Protocols, Link Layer Addressing, Ethernet, Interconnections: Hubs and Switches, PPP: The Point-to-Point Protocol, Link Virtualization

UNIT II:

Routing and Internetworking: Network–Layer Routing, Least-Cost-Path algorithms, Non-Least-Cost-Path algorithms, Intradomain Routing Protocols, Interdomain Routing Protocols, Congestion Control at Network Layer. **Logical Addressing:** IPv4 Addresses, IPv6 Addresses - **Internet Protocol:** Internetworking, IPv6, Transition from IPv4 to IPv6 – **Multicasting Techniques and Protocols:** Basic Definitions and Techniques, Intradomain Multicast Protocols, Interdomain Multicast Protocols, Node-Level Multicast algorithms

Transport and End-to-End Protocols: Transport Layer, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Mobile Transport Protocols, TCP Congestion Control **Application Layer:** Principles of Network Applications, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, Domain Name System (DNS), P2P File Sharing, Socket Programming with TCP and UDP, Building a Simple Web Server.

UNIT III:

Wireless Networks and Mobile IP: Infrastructure of Wireless Networks, Wireless LAN Technologies, IEEE 802.11 Wireless Standard, Cellular Networks, Mobile IP, Wireless Mesh Networks (WMNs)

Optical Networks and WDM Systems: Overview of Optical Networks, Basic Optical Networking Devices, Large-Scale Optical Switches, Optical Routers, Wavelength Allocation in Networks, Case Study: An All-Optical Switch

UNIT IV

VPNs, Tunneling and Overlay Networks: Virtual Private Networks (VPNs), Multiprotocol Label Switching (MPLS), Overlay Networks – **VoIP and Multimedia Networking:** Overview of IP Telephony, VoIP Signaling Protocols, Real-Time Media Transport Protocols, Distributed Multimedia

Networking, Stream Control Transmission Protocol
Mobile A-Hoc Networks: Overview of Wireless Ad-Hoc Networks, Routing in Ad-Hoc Networks, Routing Protocols for Ad-Hoc Networks – **Wireless Sensor Networks:** Sensor Networks and Protocol Structures, Communication Energy Model, Clustering Protocols, Routing Protocols

Learning Resources:

Text Books:

1. James F. Kurose, Keith W.Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3 ed, Pearson Education, 2007.
2. Nader F. Mir, *Computer and Communication Network*. Pearson Education, 2007.

Reference Books:

1. Behrouz A. Forouzan, *Data Communications and Networking*. 4 ed, TMH, 2007.
2. Greg Tomsho, Ed Tittel and David Johnson, *Guide to Networking Essentials*. 5 ed, Thomson.
3. S.Keshav, *An Engineering Approach to Computer Networking*. Pearson Education.
4. Diane Teare and Catherine Paquet, *Campus Network Design Fundamentals*. Pearson Education (CISCO Press)
5. Andrew S. Tanenbaum, *Computer Networks*. 4 ed, PHI.
6. A.Farrel and Elsevier, *The Internet and its Protocols*.

CSCS2003
EMBEDDED COMPUTING SYSTEMS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: The objective of this course is to equip the students with 8051 microcontroller programming concepts and tools needed for embedded system design. Embedded systems have become the next inevitable wave of technology, finding application in diverse fields of engineering.

- Study the Architecture of microcontroller.
- Programming the microcontroller.
- To understand the meaning of embedded system and applications in which they are used.
- To study various aspects of embedded system design from Hardware and Software points of view.
- To study various embedded design methodologies and tools

Learning Outcomes: Students will be able to

- Differentiate between microprocessor and microcontroller
- Develop microcontroller programming
- Design hardware and software for minimum microcontroller based system
- Select best suited microcontroller for specified application
- State difference between general purpose computer system and ES
- State application of ES in various fields.
- Draw hardware and software architecture of ES
- Design and implement simple embedded systems

UNIT I:

Introduction to Embedded Systems: Embedded systems; Processor embedded into a system; Embedded hardware units and devices in a system; Embedded software in a system; Examples of embedded systems; Embedded System-on-Chip (SoC) and use of VLSI circuit design technology; Complex systems design and processors; Design process in embedded system. Formalization of system design; Design process and design examples; Classification of embedded systems; Skills required for an embedded system designer.

UNIT II:

Devices: I/O types and examples; Serial communication devices; Parallel device ports;Sophisticated interfacing features in device ports. Wireless devices; Timer and counting devices; Watchdog timer; Real time clock.

Communication Buses for Device Networks : Networked embedded systems; Serial bus communication protocols; Parallel bus device protocols; Internet enabled systems; Wireless and mobile system protocols.

Device Drivers and Interrupts Service Mechanism: Device access without interrupts; ISR concept; Interrupt sources; Interrupt servicing mechanism; Multiple interrupts; Context and the periods for context-switching, interrupt latency and deadline.

UNIT III:

8051 Architecture, Memory Organizations and Real World Interfacing : 8051 Architecture; Real world interfacing, Processor and Memory Organization.

Program Modeling Concepts, Processes, Threads, and Tasks: Program models; DFG models; State machine programming models for event controlled program flow; Modeling of multiprocessor systems. Multiple processes in an application; Multiple threads in an application; Tasks and task states; Task and data; Distinctions between functions, ISRs and tasks.

UNIT IV

Real-time Operating systems: Operating System services; Process management; Timer functions; Event functions; Memory management; Device, file and I/O sub-systems management; Interrupt routines in RTOS environment and handling of interrupt source calls. **Embedded Software Development, Tools:** Introduction; Host and target machines; Linking and locating software; Getting embedded software in to the target system; Issues in hardware software design and co-design; Testing on host machine; Simulators; Laboratory tools.

Learning Resources:

Text Books:

1. Rajkamal, *Embedded Systems Architecture, Programming and Design*. 2 ed, TMH, 2008.

References:

1. Wayne Wolf, *Computers as Components Principles of Embedded Computer System Design*. Elsevier, 2005.
2. Steve Heath, *Embedded Systems Design*. 2 ed, Elsevier, 2003.
3. Dr. K.V.K.K. Prasad, *Embedded/ Real-Time Systems: Concepts, Design and Programming – The Ultimate Reference*. Dreamtech. Press, 2004.

CSCS2004
DATA MINING

Lecture	**:**	4 hrs/ Week	**Internal Assessment:**	40
Tutorial	**:**	-	**Final Examination:**	60
Practical	**:**	-	**Credits:**	4

Objectives:

- Giving the basic concepts to Data Mining and Data Warehousing.
- Provide the basic data preprocessing concepts.
- Provide the students with a complete background on Data warehousing, and Data mining basic algorithms, essential concepts, and popular techniques.
- Equip the students with sufficient knowledge so that future projects may be identified.
- Students will learn how to analyze the data, identify the problems, and choose the relevant algorithms to apply. Then, they will be able to assess the strengths and weaknesses of the algorithms and analyze their behavior on real datasets.

Learning Outcomes: **After completion of this course, the student will be able to:**

- Understand different methods of preprocessing data.
- Design and implement simple data warehouse applications.
- Design and implement simple data cubes and OLAP operations for business applications.
- Identify the problems, and choose the relevant Data Mining algorithms to apply for them.

UNIT I:

Data Mining: Introduction, Data Mining, Kinds of Data, Data Mining Functionalities, Classification of Data Mining Systems, Major issues in Data Mining.

Data Preprocessing – Data cleaning, Data Integration & Transformation, Data Reduction, Discretization & Concept Hierarchy Generation, Data Mining Primitives.

UNIT II:

Data Warehouse: Introduction, A Multi-dimensional data model, Data Warehouse Architecture, Data Warehouse Implementation.

Mining Association rules in Large Databases: Association rule mining, mining single-dimensional Boolean Association rules from Transactional Databases, Mining Multi-dimensional Association rules from relational databases & Data Warehouses.

UNIT III:

Classification and Prediction: Introduction, Classification by Decision tree induction, Bayesian Classification, Classification by Back propagation, Other Classification Methods, Prediction, Classifier accuracy.

UNIT IV

Cluster Analysis: Introduction, Types of data in Cluster analysis, A categorization of major clustering methods, partitioning methods, Hierarchical methods, Density-Based Methods: DBSCAN, Grid-based Method: STING; Model-based Clustering Method: Statistical approach, Outlier analysis.

Learning Resources:

Text Books:

1. Jiawei Han Micheline Kamber, *Data Mining Concepts and Techniques.* Morgan Kaufmann Publishers.

Reference Books:

1. Ralph Kinball, *Data Warehouse Toolkit.* John Wiley Publishers.
2. Margaret H.Dunham, *Data Mining (Introductory and Advanced Topics).* Pearson Education.
3. Sam Anahory, Dennis Murray, *Data Warehousing in the Real World – A Practical Guide for Building Decision Support Systems.* Pearson Education.
4. G.K.Gupta, *Introduction to Data Mining with Case Studies.* PHI, 2006.

CSCS2005 A
SOFTWARE TESTING METHODOLOGIES

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: The students will learn

- Basic software debugging methods.
- White box and Black box testing methods
- Writing the testing plans
- Different testing procedures for testing programs

Learning Outcomes: After completion of the course the students will be able to:

- Understand the basic testing procedures.
- Write test plans for different console and GUI applications.
- Test the applications manually and by automation by using different testing methods

UNIT I:

Principles of Testing, System and Acceptance Testing: System testing overview, Why do system testing, Functional Vs Nonfunctional testing, Functional system testing, Nonfunctional testing, Acceptance testing, summery of testing Phases.

Performance Testing: Introduction, Factors governing performance testing, Tools for performance testing, Process for performance testing, Challenges.

UNIT II:

Regression Testing: What is regression testing, Types of regression testing, When do regression testing, How to do regression testing, Best practices in regression testing.

Ad hoc Testing: Overview of Ad hoc testing, Buddy testing, Pair testing, Exploratory testing, Iterative testing, Agile and extreme testing, Defect seeding.

Testing of Object oriented Systems: Introduction, Primer on object-oriented software, Differences in OO testing.

UNIT III:

Usability and Accessibility Testing: What is usability testing, Approach to usability, when to do usability testing, How to achieve usability, Quality factors for usability, Aesthetics testing, Accessibility testing, Tools for usability, Usability lab setup, Test roles for usability.

Organization Structures for testing teams: Dimension of organization structures, Structures in single-product companies, structures for multi-product companies, Effect of globalization and geographically distributed teams on product testing, Testing services organizations, Success factors for testing organizations.

UNIT IV

Building an Effective Internal IT Audit Function: The Audit department's real mission, the concept of independence and how to avoid misusing it, how to add value beyond formal audits via consulting and early involvement, how to enhance effectiveness by building relationships, the role of IT audit and how to choose the right focus, how to build and maintain and effective IT audit team .

Audit Process: the different types of internal controls, how you should choose what to audit, how to conduct the basic stages of the audit, planning, field work and documentation, issue discovery and issue validation, solution development, report drafting and issuance, issue tracking

Learning Resources:

Text Books:

1. Srinivasan Desikan and Gopalaswamy Ramesh, *Software Testing Principles and Practices*. Pearson Education.
2. Chris Davis, Mike Schiller, and Kevin 'IT, *Auditing using Controls to Protect Information Assets*. Wheeler Tata Mc-GRAW-Hill

CSCS2005 B
PATTERN RECOGNITION

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives:

- Understand the concept of patterns and the basic approach to the development of pattern recognition algorithms
- Understand and apply methods for data preprocessing, feature extraction, and feature selection to multivariate data
- Understand and apply supervised and unsupervised classification methods to detect and characterize patterns in real-world data
- Develop prototype for pattern recognition algorithms that can be used to study algorithm behavior and performance against real-world multivariate data.

Learning Outcomes:

- The students are exposed to the underlying principles of pattern recognition and on the methods used to develop and deploy applications in the real world.
- An Emphasis is placed on the pattern recognition application development process, which includes problem identification, concept development, algorithm selection, system integration, and test and validation.
- Understand the basic concepts and methods for the recognition of patterns in data.
- Understand working knowledge of the pattern recognition application development process.

UNIT I:

Introduction: Machine perception, pattern recognition example, pattern recognition systems, the design cycle, learning and adaptation.

Bayesian Decision Theory: Introduction, continuous features – two categories classifications, minimum error-rate classification- zero–one loss function, classifiers, discriminant functions, and decision surfaces.

UNIT II:

Normal density: Univariate and multivariate density, discriminant functions for the normal density different cases, Bayes decision theory – discrete features, compound Bayesian decision theory and context.

Maximum likelihood and Bayesian parameter estimation: Introduction, maximum likelihood estimation, Bayesian estimation, Bayesian parameter estimation–Gaussian case

UNIT III:

Un-supervised learning and clustering: Introduction, mixture densities and identifiability, maximum likelihood estimates, application to normal mixtures, K-means clustering. Date description and clustering – similarity measures, criteria function for clustering.

Component analyses: Principal component analysis, non-linear component analysis; Low dimensional representations and multi dimensional scaling.

UNIT IV

Discrete Hidden Morkov Models : Introduction, Discrete–time markov process, extensions to hidden Markov models, three basic problems for HMMs

Continuous hidden Markov models : Observation densities, training and testing with continuous HMMs, types of HMMs.

Learning Resources:

Text Books:

1. Richard O. Duda, Peter E. Hart and David G. *Stroke Pattern Classifications*. 2 ed Wiley Student Edition
2. Lawerence Rabiner and Biing Hwang, *Fundamentals of Speech Recognition*. Pearson Education.

Reference Book :

1. Earl Gose, Richard John Baugh and Steve Jost, *Pattern Recognition and Image Analysis*. PHI, 2004.

CSCS2005C
NATURAL LANGUAGE PROCESSING

Lecture	**:**	4 hrs/ Week	**Internal Assessment:**	40
Tutorial	**:**	-	**Final Examination:**	60
Practical	**:**	-	**Credits:**	4

Objectives:

- Learn useful concepts, models, algorithms, and techniques.
- Learn basic knowledge of probability, formal language and automata theory, machine learning, computational, linguistic, and programming skills.
- Engineering issues involved in analysis and design natural language systems.
- Practice of the techniques used in building natural language systems.
- Appreciate the complexities of natural language.

Learning Outcomes: Upon completion of this course students will be able to

- Apply knowledge of mathematics, science, and engineering.
- Design a system, component, or process to meet desired needs within realistic constraints.
- Use the techniques, skills, and modern engineering tools necessary for engineering practice.
- Use the computer/IT tools relevant to the discipline along with an understanding of their processes and limitations.

UNIT I:

Introduction: Regular Expressions and Finite State Automata – Morphology and Finite State Transducers – Computational Phonology and Text to speech - N-grams : Counting words in Corpora – Simple N- grams – Smoothing – Entropy – HMMS

UNIT II:

Speech Recognition: Speech Recognition Architecture – Overview of HMM – Advanced Methods for decoding – Training a speech Recognizer – Human Speech Recognition – Part of Speech Tagging – Rule Based, Stochastic Part-of-Speech Tagging – Transformation Based Tagging.

UNIT III:

Context Free Grammars for English: Context Free Rules and Trees, Sentence Level Constructions, Coordination, Agreement, Grammars and Human Processing, Parsing with Context Free Grammars, Top down Parser – Problems with Basic Top Down Parser – Finite State Parsing Methods - Representing Meaning: Computational Desiderata for Representations – Meaning Structure of Language – First Order Predicate Calculus - Semantic Analysis: Syntax driven Semantic Analysis – Attached for a Fragment of English- Integrating Semantic Analysis into the Early Parser, Robust Semantic Analysis - Dialogue and Machine

UNIT IV

Translation: Dialogue Acts – Automatic, Plan inferential, Cue based Interpretation of Dialogue Acts – Dialogue Structure and coherences – Dialogue Managers - Language Similarities and Differences – The Transfer Metaphor – The Interlingua Idea- Direct Translation – Using Statistical Techniques – Usability and System Development

Learning Resources:

Text Books:

1. D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Pearson Education, 2004.

Reference Books:

1. C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Massachusetts Institute of Technology, 2003
2. James Allen, *Natural Language Understanding*. The Benajmins/Cummings Publishing Company Inc. 1994. ISBN 0-8053-0334-

CSCS2005D
HUMAN COMPUTER INTERACTION

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: **After studying the course the student can be able to:**

- Design, implement and evaluate effective and usable graphical computer interfaces.
- Describe and apply core theories, models and methodologies from the field of HCI.
- Describe and discuss current research in the field of HCI.
- Describe special considerations in designing user interfaces

Learning Outcomes: **Upon completion of this course the student will be able to:**

- Understands the user-centered design cycle and how to practice this approach to design your own website or other interactive software systems
- critique existing website and other interactive software using guidelines from human factor theories
- analyze one after another the main features of a GUI: the use of colors, organization and layout of content, filling the interface with useful and relevant information, and communication techniques; and to critique designs in order to provide better solutions
- Evaluate a GUI prototype using a questionnaire

UNIT I:

Introduction: Importance of user interface, definition, importance of good design, A brief history of Screen Design

Graphical User interface: Popularity of graphics, the concept of direct manipulation, graphical system, characteristics, Web user – interface popularity, characteristics- principles of user interface

UNIT II:

Design Process: Human interaction with computers, importance of human characteristics, human considerations, human interaction speeds

Screen designing: Interface design goals, screen meaning and purpose, organizing screen elements, ordering of screen data and content, screen navigation and flow, visually pleasing composition, amount of information, focus and emphasis, presenting information simply and meaningfully, technological considerations in interface design.

UNIT III:

Windows: Characteristics, components, operations. Selection of device based and screen based controls.

Components: **Icons and images, Multimedia, choosing proper colors**

UNIT IV

Software Tools: Specification methods, interface, Building tools

Interaction devices: Keyboard and function keys, pointing devices, speech recognition, digitization and generation, image and video displays, drivers.

Learning Resources:

Text Books:

1. Wilbert O Galitz, *The Essential Guide to UserIinterface Design*, 2 ed, Wiley DreamaTech
2. Ben Shneidermann, *Designing the User Interface*. 3 ed, Pearson Education Asia

Reference Books:

1. Alan Dix, Janet Fincay, Gre Goryd, Abowd and Russell Bealg, *Human Computer Interaction*. Pearson.
2. Prece, Rogers and Sharps, *Interaction Design*. Wiley Dreamatech.
3. Soren Lauesen, *User Interface Design*. Pearson Education.

CSCS2006A
CRYPTOGRAPHY AND NETWORK SECURITY

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: Students Learn:

- Principles and practice of network security.
- Techniques of network security
- Fundamental aspects of security in a modern networked environment
- Basic cryptographic techniques algorithms and protocols
- Computational issues in implementing cryptographic protocols and algorithms.

Learning Outcomes: By the end of this course, students will be able to:

- Understand the basic principles and terminology in network security
- Identify the possible threats to each mechanism and ways to protect against these threats.
- Understand the requirements of real-time communication security.
- Implement cryptographic protocols and algorithms.

UNIT I:
Security attacks, A model for network security; Classical techniques: Encryption, Steganography; Modern techniques: simple DES, Block cipher principles, Differential and linear cryptanalysis; Triple DES, RC5, Blowfish, CAST-128, RC2; Characteristics of Advanced Symmetric block ciphers.

UNIT II:
Conventional Encryption: Placement of Encryption function, Traffic confidentiality, key distribution, Random number generation. Public key cryptography: principles, RSA algorithm, key management, Diffie-Hellman key exchange, Elliptic curve cryptography

UNIT III:
Number Theory: Prime and relative prime numbers, modular arithmetic: theorems, testing for primality, Euclid's algorithm, Chinese remainder theorem, discrete logarithms.
Message Authentication and Hash functions: Authentication requirements and functions, security of hash functions and MACs. Message digest algorithm, secure hash algorithm.
Digital signatures and standards.

UNIT IV
Kerberos authentication and PGP email security; IP Security: overview, architecture, authentication, encapsulating security payload, key management; Web Security: Web Security requirements, Secure sockets layer and Transport layer security, Secure Electronic Transaction. Intruders, Viruses and Worms: Intruders, Viruses and Related threats. Fire Walls: Fire wall Design Principles, Trusted systems.

Learning Resources:

Text Book:

1. William Stalligs, *Cryptography and Network Security: Principles and Practice.* 4 ed, Pearson Education.
2. Mark Burgess, *Principles of Network and Systems Administration.* John Wiley.

Reference Books:

1. Charlie Caufman, Radia Perlman and Mike Speciner, *Network Security- Private Communication in a Public World.* 2 ed, PHI.

CSCS2006B
SECURED DATABASE APPLICATIONS DEVELOPMENT

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: Student Learns

- Providing security to database
- Various Authentication methods
- Defining and implementing password policies
- To Implement a Virtual Private Databases
- Database Auditing Models
- Auditing Projects
- To develop a secured authentication repository

Learning Outcomes: Upon completion of the course the student will be able to

- Understand various application security models and encryption of data.
- provide security to Email
- Implement a VPD using views
- Implement oracle VPD and Viewing VPD policies
- Know about Auditing Database Activities
- Develop a online database
- Track database changes

UNIT I:

Security Architecture: Introduction, Security, Information Systems, Database management systems, Information security, Information security Architecture, database security, Asset types and their value, Security methods.

Operating System Security Fundamentals: Introduction, operating systems overview, security environment, components, Authentication methods, user administration, password policies, Vulnerabilities of operating systems, E- Mail security.

UNIT II:

Administration of Users: Introduction, user authentication, operating system authentication, creating/removing/modifying users, default/remote users, Database links, Linked servers, remote servers. Profiles, Password Policies, Privileges, and Roles.

Introduction, Defining and using profiles, Designing and implementing password policies, Granting and revoking user privileges, creating, Assigning and revoking user roles.

UNIT III:

Database Application Security Models : Introduction, Types of users, security models, application types, application security models and Data encryption.

Virtual Private Databases (VPD): Introduction, Overview, implementing a VPD using views and application context. Implementing oracle VPD, Viewing VPD policies and application context using: data dictionary, policy manager, implementing row and column level security with SQL server.

UNIT IV

Database Auditing Models, Application Data Auditing: Database Auditing Models: Introduction, Auditing overview, environment, process, objectives, classification and types, benefits and side effects of auditing.

Application Data Auditing: Introduction, DML auction auditing architecture. Triggers, fine grained auditing, DML statement audit trail and auditing application errors with Oracle.
Auditing Database Activities, Security and Auditing Project Cases:
Auditing Database Activities: Introduction, usage of database activities, creating DLL triggers, auditing database activities with oracle. Security and Auditing project cases: Introduction, case study for developing an online database, taking care of payroll, tracking database changes and developing a secured authentication repository

Learning Resources:

Text Books:

1. Hassan Afyouni, *Database Security and Auditing.* Cengage Learning, 2007.

Reference Books:

1. S. Castano, M. Fugini, G. Martella and P. Samarati, *Database Security.* Addison-Wesley, 1994.
2. Ron Ben Natan, *Implementing Database Security and Auditing.* Elsevier, Indian Reprint, 2006
3. Clark, Holloway and List *The security Audit and Control of Databases.* UK: Ash gate.
4. Douglas, *Security and Audit of Database System.* UK: Blackwell
5. Fernandez, Summers and Wood, *Database Security and Integrity,* Addison Wesley

CSCS2006C
BIOINFORMATICS

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: Students learn

- Computational approach to solve biological problems.
- Different data visualization techniques
- Basic biological databases and algorithms for proteomics and genomics analysis.
- Bioinformatics packages to solve the biological problems.

Learning Outcomes: By the end of this course, the students will be able to

- Know the differences between genomics and proteomics.
- Understand different data visualization techniques
- Understand and analyze how to solve the biological problems using computational approach
- Understand and analyze internet packages of bioinformatics.

UNIT I:

The Central Dogma: The Killer Application, Parallel Universes, Watson's Definition, Top Down Versus Bottom up, Information Flow, Convergence Databases, Data Management, Data Life Cycle , Database Technology ,Interfaces , Implementation

Networks: Networks, Geographical Scope, Communication Models, Transmissions Technology, Protocols, Bandwidth, Topology, Hardware, Contents, Security, Ownership, Implementation, Management.

UNIT II:

The Search Process: Search Engine Technology, Searching and Information Theory, Computational methods , Search Engines and Knowledge Management

Data Visualization: Data Visualization, sequence visualization, structure visualization, user Interface, Animation Versus simulation, General Purpose Technologies.

UNIT III:

Statistical Concepts: Micro-arrays, Imperfect Data, Randomness, Variability, Approximation, Interface Noise, Assumptions, Sampling and Distributions, Hypothesis Testing, Quantifying Randomness, Data Analysis, Tool selection statistics of Alignment.

Data Mining: Clustering and Classification , Data Mining , Methods , Selection and Sampling , Preprocessing and Cleaning , Transformation and Reduction , Data Mining Methods , Evaluation , Visualization , Designing new queries ,Pattern Recognition and Discovery , Machine Learning , Text Mining , Tools.

UNIT □ IV

Pairwise sequence alignment, Local versus global alignment, Multiple sequence alignment, Computational methods , Dot Matrix analysis , Substitution matrices , Dynamic Programming, Word methods, Bayesian methods, Multiple sequence alignment, Dynamic Programming, Progressive strategies, Iterative strategies, Tools, Nucleotide Pattern Matching, Polypeptide pattern matching , Utilities , Sequence Databases.

Modeling and Simulation : Drug Discovery, components, process, Perspectives, Numeric considerations, Algorithms, Hardware, Issues, Protein structure, Ab-Initio Methods, Heuristic methods, Systems Biology, Tools, Collaboration and Communications, Standards, Issues, Security, Intellectual property.

Learning Resources:

Text Books:

1. Bryan Bergeron, *Bio Informatics Computing.* PHI, 2003.
2. Lacroix and Terence Critchlow, *Bio Informatics, Managing Scientific Data.* Elsevier

Reference Books

1. Attwood and Smith, *Introduction to Bio Informatics.* Longman, 1999.
2. D Srinivasa Rao, *Bio-Informatics.* Biotech.
3. Rastogi and Mendiratta, *Bio Informatics Methods and Applications.* PHI

CSCS2006D
SOFT COMPUTING

Lecture :	4 hrs/ Week	**Internal Assessment:**	40
Tutorial :	-	**Final Examination:**	60
Practical :	-	**Credits:**	4

Objectives: Students Learn

- General concepts and techniques in soft computing
- Soft computing techniques to practical problems
- Popular tools for Soft Computing;

Learning Outcomes: **Upon completion of this course the student will be able to:**

- Understand the need and usage of Soft Computing in various areas
- Know the steps involved in the development of Soft Computing;
- Use popular tools for Soft Computing;
- Design and implement computing systems by using appropriate Soft Computing techniques and tools.

UNIT I:

Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen's self organizing networks - Hopfield network.

UNIT II:

Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition - Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.
Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees - Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls - Simulated annealing – Evolutionary computation.

UNIT III:

Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction - Rank method - Rank space method.

UNIT IV

Soft computing and Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks - Frames - Objects - Hybrid models - Applications.

Learning Resources:

Text Books:

1. Jang J.S.R., Sun C.T. and Mizutani E, *Neuro-Fuzzy and Soft Computing*. PHI/Pearson Education, 1998.

Reference Books:

1. Timothy J.Ross, *Fuzzy Logic with Engineering Applications*. McGraw Hill, 1997.
2. Laurene Fausett, *Fundamentals of Neural Networks*. Pearson Education/PHI 1994.
3. George J. Klir and Bo Yuan, *Fuzzy sets and Fuzzy Logic*. PHI/Pearson Education.
4. Nih J.Nelsson, *Artificial Intelligence - A New Synthesis*. Harcourt Asia Ltd., 1998.
5. D.E.Goldberg, *Genetic Algorithms: Search, Optimization and Machine Learning. NY:* Pearson Education, 1989.

CSCS2051
SOFTWARE ENGINEERING LAB

Lecture :	-	**Internal Assessment:**	25
Tutorial :	-	**Final Examination:**	50
Practical :	3 hrs/ Week	**Credits:**	2

Objectives:

- Introduce the lab environment and tools used in the software engineering lab: WebCT and SynchEye.
- Become familiar with a configuration management case tool (Microsoft Visual Source Safe).
- Learn how to prepare project plans
- Study the benefits of visual modeling.
- Learn use case diagrams: discovering actors and discovering use cases.
- Practice use cases diagrams using Rational Rose.
- Data model: entity-relationship diagram (ERD).
- Functional model: data flow diagram (DFD).
- Documenting Use Cases and Activity Diagrams
- Object-Oriented Analysis: Discovering Classes
- Interaction Diagrams: Sequence & Collaboration Diagrams

Learning Outcomes: Upon completion of the lab course, the student will be able to

- Plan software project development life cycle
- Write System Requirements Speciation (SRS) for given problem
- Document use case, activity, DFDs, Sequence and Collaboration Diagrams for given problem.

Week 1:

Use Visual Source Safe (VSS) tool to manage your documents for the following Tasks

- Create a project and add some java files to it (at least 3 files).
- Label the existing files.
- Check out all the files and modify one of them.
- Check the edited file back in.
- View the revision history of the edited file and show the differences between the old and the new versions.
- Search for any unchecked in files.

Week 2:

Use MS Project 2002 to create a series of tasks leading to completion of a project of your choice. For your project, you need to:

- Set start or ending dates.
- Develop a list of tasks that need to be completed.
- Establish any sub tasks and create links.
- Create any links between major tasks.
- Assign a specific amount time for each task.
- Assign resources for each task.
- Create task information for each item you put into the list.

Week 3:

Read carefully the following problem statement

We are after a system that controls a recycling machine for returnable bottles and cans. The machine will allow a customer to return bottles or cans on the same occasion.

When the customer returns an item, the system will check what type has been returned. The system will register how many items each customer returns and, when the customer asks for a receipt, the system will print out what he deposited, the value of the returned items and the total return sum that will be paid to the customer.

The system is also be used by an operator. The operator wants to know how many items of each type have been returned during the day. At the end of the day, the operator asks for a printout of the total number of items that have been deposited in the machine on that particular day. The operator should also be able to change information in the system, such as the deposit values of the items. If something is amiss, for example if a can gets stuck or if the receipt roll is finished, the operator will be called by a special alarm signal.

After reading the above problem statement, find:

1. **Actors**
2. **Use cases** with each actor
3. Find **extended** or **uses** use cases (if applicable)
4. Finally : draw the main **use case diagram**:

Week 4:

a. Create an Entity Relationship Diagram (ERD) for an airline reservation system.

b. Create a DFD for:
   i. Student Registration System.
   ii. (a + b) * (c + a * d)

Week 5:

1. Take two of the use cases from recycling machine problem (Lab 5) and write flow of events for those. You should work in groups of three to solve this problem.
2. Practice activity diagram from the example in PPT (Lab 7) for course catalog creation.

Week 6:

Consider the following requirements for the Video Store system. Identify the candidate classes:
The video store keeps in stock an extensive library of current and popular movie titles. A particular movie may be held on video tape or disk.
Video tapes are in either "Beta" or "VHS" format. Video disks are in DVD format. Each movie has a particular rental period (expressed in days), with a rental charge to that period. The video store must be able to immediately answer any inquiries about a movie's stock availability and how many tapes and/or disks are available for rental. The current condition of each tape and disk must be known and recorded.
The rental charge differs depending on video medium: tape or disk (but it is the same for the two categories of tapes: Beta and VHS).
The system should accommodate future video storage formats in addition to VHS tapes, Beta tapes and DVD disks. The employees frequently use a movie code, instead of movie title, to identify the movie. The same movie title may have more than one release by different directors.

Week 7:

We all have used an elevator. The following steps describe the scenario of what happens at the elevator door from outside.

1. The passenger presses the button of either up or down depending on where he wants to go.
2. Then he will see that the button he pressed is illuminated.
3. The elevator is now moving to his floor.
4. When the elevator reached his floor it stops.
5. Now the button which was illuminated now off.
6. The door opens and the passenger enters.
7. The door closes.

The objects:

3. **Passenger** (he is the actor).
4. **Floor button** (this is the interface class, the actor interacts with this object).
5. **Elevator controller** (this the control class, which coordinates the activities of the scenario).
6. **Elevator** (the entity class, which represents the machine itself which moves up and down).
7. **Door** (another entity class, which represents the door which opens and closes).

Week 8:

Refer to Lab 8 exercises; assume that the Video Store needs to know if a video tape is a brand new tape or it was already taped over (this can be captured by an attribute is_taped_over); assume also that the storage capacity of a video disk allows holding multiple versions of the same movie, each in a different language or with different endings.

Use the identified classes from Lab 8 and find their attributes, operations and the relationships between the classes (build the UML diagram).

Week 9:

Here is what happens in a microwave oven:

1. The oven is initially in an **idle state with door open**, where the light is turned on.
2. When the door is closed it is now in **idle with door closed**, but the light is turned off.
3. If a button is pressed, then it moves to **initial cooking stage**, where the timer is set and lights are on, and heating starts
4. At any moment the door may be opened, the **cooking is interrupted,** the timer is cleared, and heating stops.
5. Also while cooking, another button can be pushed and **extended cooking state** starts, where the timer gets more minutes. At any moment door can be opened here also.
6. If the time times out, then **cooking is complete**, heating stops, lights are off, and it sounds a beep.
7. When the door is open, again the oven is in idle state with the door open.

Draw a state transition diagram for the microwave oven

Week 10:

Think about any system and its components and draw deployment and component diagrams.

Week 11:

A tutorial on how to use JUnit with some example will be presented. Please refer to Lab 13 slides for an overview of software testing and JUnit tutorial with some examples.

Write a JUnit test class for testing

```
public class Exercise {
   /** Return the minimum of x and y. */
    public static int min(int x, int y) { ... }
}
```

Learning Resources:

Text Books:

1. Roger S Pressmen, *Software Engineering – A practitioner's Approach*.6 ed, TMH
2. James Rumbaugh, Jacobson and Booch, *Unified Modeling Language Reference Manual.* Pearson Education.

References:

1. Visual Source Safe (VSS) tool Manual
2. MS Project Manual
3. IBM Rational Rose Manual

CSCS2052
DATA MINING LAB

Lecture :	-	**Internal Assessment:**	25
Tutorial :	-	**Final Examination:**	50
Practical :	3 hrs/ Week	**Credits:**	2

Objectives: After studying the course the student can be able to:

- Perform various data preprocessing operations
- Perform various data cube operations
- Analyze and visualize large Databases
- Perform classification task with supervised and unsupervised learning

Learning Outcomes: Upon completion of this course the student will be able to:

- Define and construct Data Warehouses for Large Databases
- Compute Data Summaries and Reports
- Classification and Clustering of Data Objects
- Apply Data Mining on Various Applications

List of Experiments

1. Preprocessing: Removal specified attribute, discrimination of a continuous valued attribute, standardization and normalization of data.
2. Preprocessing: Selecting the features subset using different attribute selection measures.
3. Data Warehousing: Cube operations- Drill down, drill up, slice and rotate operations.
4. Association Mining: Finding Association Rules using Apriori principle for Bank dataset
5. Association Mining: Finding Association Rules using Apriori principle for AllElectronics Transaction Dataset.
6. Classification: classify the AllElectronics dataset records using Decision Tree based classification model.
7. Classification: classify the AllElectronics dataset records using Rule-Based classification model.
8. Classification: classify the AllElectronics dataset records using Naïve Bayes classification model.
9. Classification: classify the AllElectronics dataset records using Multilayer Feed forward Network classification model.
10. Clustering: Use k-means clustering technique to classify the given dataset.
11. Clustering: Use Hierarchical based clustering technique to classify the given dataset
12. Clustering: Use DBScan clustering technique to classify the given dataset

Learning Resources:

Text Books:

1. Jiawei Han and Micheline Kamber, *Data Mining Concepts and Techniques.* Morgan Kaufmann Publishers.

Reference Books:

1. Ralph Kinball, *Data Warehouse Toolkit.* John Wiley Publishers.
2. G.K.Gupta, *Introduction to Data Mining with Case Studies.* PHI Publications, 2006

Web References:
www.cs.waikato.ac.nz/ml/**weka**

CSCS4051

PROJECT

Credits : 24 **Internal Assessment : 100** **External Assessment:200**