CHRIST COLLEGE (AUTONOMOUS), IRINJALAKUDA

IRINJALAKUDA, THRISSUR - PIN 680 125

DEGREE OF
MASTER OF SCIENCE
(CHOICE BASED CREDIT AND SEMESTERSYSTEM)

UNDER THE

FACULTY OF SCIENCE

SYLLABUS

(FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2017 ONWARDS)

BOARD OF STUDIES IN COMPUTER SCIENCE (PG)

IRINJALAKUDA, THRISSUR - PIN
680 125 KERALA, 673 635, INDIA
JULY, 2017

CONTENTS

REGULATIONS FOR THE DEGREE OF
MASTER OF SCIENCE
(COMPUTER SCIENCE)

EFFECTIVE FROM THE ACADEMIC YEAR 2014 - 15

1 PROGRAMME OBJECTIVES

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

a) To equip students to take up challenging research oriented responsibilities and courses for their higher studies/profession.

b) o train and equip the students to meet the requirements of the Software industry in the country and outside.

c) To motivate and support the students to prepare and qualify challenging competitive examinations such as JRF/NET/JAM/GATE etc.

2 GENERAL PROGRAMME STRUCTURE

Duration: The duration of the MSc (Computer Science) programme shall be 4 semesters distributed over a period of 2 academic years. The odd semesters (1, 3, 5) shall be from June to October and the even Semesters (2, 4, 6) shall be from November to March. Each semester shall have 90 working days inclusive of all examinations.

Courses: The programme includes four types of courses, viz., Core courses (Code C) and Elective Courses (Code E). Core courses are of theory and practical oriented. There is a Project Work which is to be undertaken by all students. No course shall have more than 4 credits. For project work and General Viva-Voce, the maximum credits shall be 8. General Viva-Voce covers questions from all courses in the programme.

Attendance: A student shall be permitted to appear for the semester examination, only if (s)he secures not less than 75% attendance in each semester. Attendance shall be maintained by the concerned department. Condonation of shortage of attendance to a maximum of 9 days in a semester subject to a maximum of two times during the whole period of the MSc Computer Science programme may be granted by the University. Benefits of attendance may be granted to students who attend the approved activities of college/university with prior concurrence of the head of the institution. Participation in such activities may be treated as presence in lieu of their absence on production of participation / attendance certificate in curricular / extracurricular activities. It should be limited to 9 days in a semester.

The condonation of shortage of attendance shall be granted according to the existing prescribed norms.

If a student registered in first semester of the MSc Computer Science programme is continuously absent from the classes for more than 14 working days at the beginning of the semester without informing the authorities the matter shall immediately be brought to the notice of the Registrar of the University. The names of such students shall be removed from the rolls.

Admission to repeat courses should be within the sanctioned strength. However if more candidates are there, the candidates who have suffered serious health problems, on production of a medical certificate issued by a physician not below the rank of a Civil Surgeon in Government service, may be permitted to repeat the course, with a written order issued by the Registrar, Calicut University (by considering his / her SGPA / CGPA and percentage of attendance). The number of such candidates should not exceed two.

Project: Every student of the MSc Computer Science programme shall have to work on a project of not less than 8 credits under the supervision of a faculty member as per the curriculum.

3 ADMISSION

The admission to all programmes will be as per the rules and regulations of the University. The eligibility criteria for admission shall be as announced by the University from time to time.

Separate rank lists shall be drawn up for reserved seats as per the existing rules. The college shall make available to all students admitted a prospectus listing all the courses offered including electives in various departments during a particular semester. The information provided shall contain title of the course and credits of the course.

There shall be a uniform calendar prepared by the University for the registration, conduct /schedule of the courses, examinations and publication of results. The University shall ensure that the calendar is strictly followed.

There shall be provision for inter collegiate and inter university transfer in third semester within a period of two weeks from the date of commencement of the semester. There shall be provision for credit transfer subject to the conditions specified by the Board of Studies concerned.

4 REGISTRATION

A student shall be permitted to register for the programme at the time of admission. A student shall be permitted to register for the examination also. If registration for

examination is not possible owing to shortage of attendance beyond condonation limit, the student shall be permitted to move to the next semester. In such cases, a request from the student may be forwarded through the principal of the college to the University within two weeks of the commencement of that semester. An undertaking from the Principal may also be obtained stating that the students will be permitted to make up the shortage of attendance in that semester after completing 6 semesters (Students shall make up the shortage of attendance in 'Repeat Semester' after completion of the programme).

The 'Repeat Semester' shall be possible only once for the entire programme and shall be done in the same college. A student who registered for the course shall successfully complete the programme within 4 years from the year of first registration. If not, such candidate has to cancel the existing registration and join afresh as a new candidate. The students who have attendance within the limit prescribed, but could not register for the examination have to apply for the token registration, within two weeks of the commencement of the next semester.

5 COURSE EVALUATION

The evaluation scheme for each course shall contain two parts: (a) internal evaluation and (b) external evaluation. 25% weight shall be given to internal evaluation and the remaining 75% to external evaluation. Therefore the ratio of weight between internal and external is 1:3. Both internal and external evaluation shall be carried out using Direct grading system.

INTERNAL EVALUATION

The internal evaluation shall be based on predetermined transparent system involving periodic written tests, assignments, seminars and attendance in respect of theory courses and based on lab tests, lab skill/records/viva and attendance in respect of practical courses.

THEORY PAPERS

The weightage assigned to various components for internal evaluation for theory papers is as shown below.

Components	Weightage
Test papers with at least 25% questions based on problems or programs (minimum two)	2
Assignments (minimum two) such as homework, problem solving, group discussions, quiz, literature survey, term-project, software exercises, etc.	1
Regularity in the class	1
Seminar	1
Total	**5**

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

The course teacher shall maintain the academic record of each student registered for the course, which shall be forwarded to the University, through the college Principal.

PRACTICAL PAPERS

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

Components	Weightage
Rough record for each experiment	1
Performance in the laboratory – coding, results	1
Fair Record	1
Regularity	1
End-semester test	1
Total	**5**

Note:

1. All students should have a rough record (observation note book) in which they write all the works to be carried out in the lab prior to his/her entering the lab. (S)he may also note down the i/p and o/p that (s)he gives for program verification in the observation note book (rough record).

2. All lab works should be neatly recorded in a Laboratory Record Book (Fair Record) in written form. However program results can be pasted in the left hand side of the fare record.

3. Chairperson, Board of Examination (PG) has to prepare the modalities of the practical papers (list of experiments to be done, number of minimum experiments required in the practical record etc) and distributed to all departments concerned, at the beginning of each semester itself. Model lists of experiments are provided with the syllabus for each practical session.

4. No candidate will be permitted to attend the end-semester test unless he/she produces certified record of the laboratory.

5. Full credit for regularity in the class can be given only if the candidate has secured minimum 90% attendance in the course. Attendance evaluation for each course is as follows:

Percentage of Attendance	Weightage
90% and above	4
85 to 89.9%	3
80 to 84.9%	2
75 to 79.9%	1
Below 75 %	0

EVALUATION COMMITTEE (EC)

For the evaluation of the Project Work (CSS4C02) and Term Paper (CSS2P07), an evaluation committee is to be constituted. One faculty is to be designated as the Course Coordinator for these courses. Committee is to be constituted by the head of the department (HOD) and (s)he shall be the Chairperson of the committee. In addition to the HOD, the Course Coordinator and at least three faculty members can be designated as the members of the committee. In case HOD is unable to represent himself/herself in the committee, (s)he can nominate a faculty in lieu for him/her as a member and the chairperson of the committee. In addition to this, faculty guiding a particular student will also be a member of the committee. At least one member of the committee should be a lady, if lady faculties are available in the department concerned. The Coordinator has to set the schedule for presentation and submission of the reports. While calculating the final score, 25% weight is to be given for the scores awarded by the guide to the student and the rest 75% weight is to be given for the average of the scores awarded to the student by remaining committee members.

TERM PAPER

A tentative list of the components for evaluation of Term Paper is as shown below. Evaluation committee can decide about the actual composition of the components and scores to be awarded for each component.

Component
Relevance of the Topic, Statement of Objectives, Correctness
Quality of Literature Survey / Product Review
Methodology / tools Adopted
Quality of Contributions
Quality of Implementation / Simulation
Quality of Testing
Identification of Future Work
Quality of the Term Paper Report
Publications/Presentations/Communications out of the Term Paper
Quality of Presentation

PROJECT WORK

Total weightage for Project Work (and General Viva Voce) shall be 72 (36 x 2). Hence the total grade points shall be 288 (72 x 4). Scheme to award internal continuous assessment grades for project work should be as follows:

Components	Weightage
Monthly progress	4
Regularity	1
Total	**5**

Regularity is to be reported by the guide to the EC, considering factors such as students' commitment to work, timely submission of assignments, punctuality and availability.

Monthly progress can be conveniently evaluated in various phases such as Formulation of Project Problem, Analysis, Design, Implementation and Testing. In each of these phases, students can be asked to make presentations of their work and submit interim reports for each phase. Components for evaluation of monthly progress is as shown below.

Component	Grade Points
Relevance of the Topic, Statement of Objectives, Methodology	20
Quality of Literature Survey/Product Review	20
Quality of Analysis Phase	20
Quality of Design Phase	20
Quality of Implementation/Simulation	50
Quality of Testing/Result Analysis	20
Quality of Contributions	20
Identification of Future Work	8
Quality of Project Report	50
Publications/Presentations out of the Project Work*	10
Quality of Presentation	15
Demonstration of the Project Work	10
General Viva Voce	25
Total	**288**
Grade is calculated by dividing total number of points obtained by a student by 72	

*In case at least one student of the batch has a publication/presentation out of his/her project work in a workshop/conference/journal/IT fest etc, this score is to be awarded for the student; no other students will deserve score for this component! If none of the students in the batch could make such an edge, then the score for this component is to be added with the component "Identification of Future Work".

The Evaluation Committee can decide about the components for monthly evaluation from the above list. See Appendix B for a sample evaluation.

EXTERNAL EVALUATION

- End semester examinations for theory and practical courses will be conducted by the University. For practical courses, end semester examinations will be conducted in even semesters.
- Evaluation for the following courses will be done internally by the concerned departments:
    - CSS2P07 Term Paper
    - CSS4E01 Elective IV
    - CSS4C01 Principles of Software Engineering
- The external examinations in theory and practical courses (excluding three courses mentioned above) are to be conducted by the University with question papers set by external experts.
- External project evaluation shall be conducted at the end of the fourth semester.
- The evaluation of the answer scripts shall be done by examiners based on a well- defined scheme of valuation. The external evaluation shall be done immediately after the examination preferably in a Centralized Valuation Camp.
- Practical examinations, project evaluation and General Viva-Voce shall be conducted by two external examiners. General Viva-Voce covers questions from all courses in the programme.
- For Project Work, if the performance of the student is below the expected benchmark (E grade), student will be given a chance to reappear within six weeks (from the date of evaluation) to present the work again, after incorporating the changes suggested by the examiners. Examiners have to submit their suggestions in writing to Chairperson, Board of Examinations PG and the concerned HOD on the day of examination itself. HOD has to convey the matter ASAP to the students concerned. The Chairperson, Board of Examinations has to inform the concerned HOD about the schedule for resubmission and revised evaluation within seven days of the date of evaluation. While submitting the revised report, the student has to produce a certificate (signed by the student, the guide and the HOD) stating that the changes suggested by the examiners are incorporated in the revised report. Also a summary of the changes made in the revised report as per the suggestions of the examiners is to be submitted (as a separate manuscript) with the revised report. If the result of the second evaluation is worth E grade, (s)he will have to appear for the end semester examinations along with regular students. This provision is only applicable for Project Work evaluation.
- Failed or improvement candidates will have to appear for the end semester examinations along with regular students.

REVALUATION

- Photocopies of the answer scripts of the end semester examinations shall be made available to the students for scrutiny on request and revaluation/scrutiny of answer scripts shall be done as per the existing rules prevailing in the University.
- Awarding of a higher grade after revaluation may be given only after a second revaluation.

IMPROVEMENT/SUPPLEMENTARY

A maximum of two courses (Core or Elective) can be improved in each semester. Improvement of a particular semester can be done only once. The student shall avail the improvement chance in the succeeding year after the successful completion of the semester concerned. The internal marks already obtained will be carried forward to determine the grades/marks in the improvement examination. If the candidate fails to appear for the improvement examination after registration, or if there is no change in the results of the improvement examination appeared, the marks/grades obtained in the first appearance will be retained.

Improvement and supplementary examinations cannot be donesimultaneously.

7 PATTERN OF QUESTION PAPERS

Duration of End Semester examinations for both theory and practical courses shall be 3 hours.

QUESTION PAPERS - THEORY

Section	No of Questions		Weightage for each question	Total
	To be Asked	To be Answered		
A: Short answer questions+	12	12	1	12
B: Short Essay	9	6	2	12
C: Essays*	6	3	4	12
			Total	**36**

+MCQ / fill in the blank / matching /one word / etc. Each question is to be answered in 7 minutes duration and should extract the critical knowledge acquired by the candidate in the subject.

* Programs / Psuecode / Problems / Derivations / Narrations. A question can have subdivisions. Each question is to be answered in 30 minutes. May be asked as a single question or in parts.

QUESTION PAPERS - PRACTICAL

Mark distribution for practical courses shall be as follows.

Component	Weightage
Algorithm/Flow diagram/UI diagram/Class Diagram	1
Implementation	1
Result/Output	1
Record	1
Viva	1
Total	**5**

PROJECT WORK

Total weightage for Project Work (and General Viva Voce) shall be 72 (36 x 2). Hence the total grade points shall be 288 (72 x 4).

The scheme of evaluation for project work shall be:

Component	Grade Points
Relevance of the Topic, Statement of Objectives, Methodology	20
Quality of Literature Survey/Product Review	20
Quality of Analysis Phase	20
Quality of Design Phase	20
Quality of Implementation/Simulation	50
Quality of Testing/Result Analysis	20
Quality of Contributions	20
Identification of Future Work	8
Quality of Project Report	50
Publications/Presentations out of the Project Work*	10
Quality of Presentation	15
Demonstration of the Project Work	10
General Viva Voce	25
Total	**288**
Grade is calculated by dividing total number of points obtained by a student by 72	

*In case at least one student of the batch has a publication/presentation out of his/her project work in a workshop/conference/journal/IT fest etc, this score is to be awarded for the student; no other students will deserve score for this component! If none of the students in the batch could make such an edge, then the score for this component is to be added with the component "Identification of Future Work".

8 CREDIT SYSTEM

Each course shall have a specific credit (whole number) depending on the academic load and the nature and importance of the course. The credit associated with each course is as listed in the prescribed scheme and syllabi.

Direct Grading System based on a 5 point scale is used to evaluate the performance (External and Internal Examination of students).

a) One Credit is equivalent to 4 periods of 60 minutes each, for theory and practical.

b) Total credits of the MSc Computer Science Programme shall be 87. The following is the semester wise credits a student must earn for the award of the degree:

Semester	Duration	Credits
I	Six Months	24
II	Six Months	25
III	Six Months	24
IV	Six Months	14
Total	**24 Months**	**87**

9 DIRECT GRADING SYSTEM

- Direct Grading System based on a 4 point scale is used to evaluate the performance (external and internal examination of students).
- Each course is evaluated by assigning marks with a letter grade (A, B, C, D, E).

Grade	Performance	Grade Point	Grade Range
A	Excellent	4	3.50 - 4.00
B	Very Good	3	2.50 - 3.49
C	Good	2	1.50 - 2.49
D	Average	1	0.50 - 1.49
E	Poor	0	0.00 - 0.49

- Each course is evaluated by assigning a letter grade (A,B,C,D or E) to that course by the method of direct grading. The internal (weightage =1) and external weightage = 3) components of a course are separately graded and then combined to get the grade of the course after taking into account of their weightage.
- An aggregate of C grade (external and internal put together) is required in each course for a pass and also for awarding the degree.
- A student who fails to secure a minimum grade for a pass in a course will be permitted to write the examination along with the next batch.
- After the successful completion of a semester, Semester Grade Point Average (SGPA) of a student in that semester is calculated using the formula given below. For the successful completion of a semester, a student should pass all courses. However, a student is permitted to move to the next semester irrespective of SGPA obtained.

- SGPA of the student in that semester is calculated using the formula

SGPA = Sum of the credit points of all courses in a semester Total credits in that semester

- The Cumulative Grade Point Average (CGPA) of the student is calculated at the end of a programme. The CGPA of a student determines the overall academic level of the student in a programme and is the criterion for ranking the students. CGPA can be calculated by the following

CGPA = Total credit points obtained in four semesters Total credits acquired

- SGPA and CGPA shall be rounded off to two decimal places. CGPA determines the broad academic level of the student in a programme and is the index for ranking students (in terms of grade points).
- An overall letter grade (Cumulative Grade) for the entire programme shall be awarded to a student depending on her/his CGPA.

10 GRADE CARDS

The University shall issue to the students grade/marks card (by online) on completion of each semester, which shall contain the following information:

i. Name of the University.
ii. Name of the college.
iii. Title of the Programme – MSc Computer Science.
iv. Semester concerned.
v. Name and Register Number of the student.
vi. Code number, Title and Credits of each course opted in the semester.
vii. Internal marks, External marks, total marks, Grade point (G) and Letter grade in each course in the semester.
viii. The total credits, total credit points and SGPA in the semester.

The Final Grade Card issued at the end of the final semester shall contain the details of all courses taken during the entire programme including those taken over and above the prescribed minimum credits for obtaining the degree. The Final Grade Card shall show the CGPA and the overall letter grade of a student for the entire programme.

11 AWARD OF DEGREE

The successful completion of all the courses prescribed for the MSc Computer Science programme with C grade shall be the minimum requirement for the award of MSc Computer Science programme degree.

12 GRIEVANCE REDRESSAL COMMITTEE

COLLEGE LEVEL

The College shall form a Grievance Redressal Committee in each department comprising of course teacher and one senior teacher as members and the HOD as Chairman. The Committee shall address all grievances relating to the internal assessment grades of the students. There shall be a college level Grievance Redressal Committee comprising of student advisor, two senior teachers and two staff council members (one shall be an elected member) as member and the Principal as the Chairperson.

UNIVERSITY LEVEL

The University shall form a Grievance Redressal Committee as per the existing norms.

13 TRANSISTORY PROVISION

Notwithstanding anything contained in these regulations, the Vice-Chancellor shall, for a period of one year from the date of coming into force of these regulations, have the power to provide by order that these regulations shall be applied to any programme with such modifications as may be necessary.

14 REPEAL

The Regulations now in force in so far as they are applicable to programmes offered by the University and to the extent they are inconsistent with these regulations are hereby repealed. In the case of any inconsistency between the existing regulations and these regulations relating to the Choice Based Credit Semester System in their application to any course offered in a College, the latter shall prevail.

MASTER OF SCIENCE COMPUTER SCIENCE

PROGRAMME STRUCTURE

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

Semester I

No	Course Code	Course Name	C	Weightage			Hrs/wk		
				I	E	T	L	P	T
1.1	CSS1C01	Discrete Mathematical Structures	4	25	75	100	4		4
1.2	CSS1C02	Advanced Data Structures	4	25	75	100	3	2	5
1.3	CSS1C03	Theory of Computation	4	25	75	100	4		4
1.4	CSS1C04	The Art of Programming Methodology	4	25	75	100	2	2	4
1.5	CSS1C05	Computer Organization and Architecture	4	25	75	100	4		4
1.6	CSS1P06	Practical I	4	25	75	100		4	4
		Total	**24**				**17**	**8**	**25**

Semester II

				Weightage			Hrs/wk		
2.1	CSS2C01	Design and Analysis of Algorithms	4	25	75	100	3	1	4
2.2	CSS2C02	Operating System Concepts	4	25	75	100	3	1	4
2.3	CSS2C03	Computer Networks	4	25	75	100	4		4
2.4	CSS2C04	Computational Intelligence	4	25	75	100	4		4
2.5	CSS2E05	Elective I	4	25	75	100	4		4
2.6	CSS2P06	Practical II	4	25	75	100		4	4
2.7	CSS2P07*	Term Paper	1	100		100		1	1
		Total	**25**				**18**	**7**	**25**

* Evaluation is to be done fully internally for this paper

List of Elective Courses (Semester II)

No	Course Code	Course Name
2.5a	CSS2E05a	Computer Graphics
2.5b	CSS2E05b	Introduction to Soft Computing
2.5c	CSS2E05c	Web Technology
2.5d	CSS2E05d	Bioinformatics
2.5e	CSS2E05e	Computer Optimization Techniques
2.5f	CSS2E05f	Numerical and Statistical Methods

Semester III

No	Course Code	Course Name	C	Weightage			Hrs/wk		
				I	E	T	L	P	T
3.1	CSS3C01	Advanced Database Management System	4	25	75	100	4	1	5
3.2	CSS3C02	Principles of Compilers	4	25	75	100	4		4
3.3	CSS3C03	Object Oriented Programming Concepts	4	25	75	100	4		4
3.4	CSS3E04	Elective II	4	25	75	100	4		4
3.5	CSS3E05	Elective III	4	25	75	100	4		4
3.6	CSS3P06	Practical III	4	25	75	100		4	4
		Total	**24**				**20**	**5**	**25**

List of Electives for CSS3E04

No	Course Code	Course Name
3.4a	CSS3E04a	Pattern Recognition
3.4b	CSS3E04b	Wireless and Mobile Networks
3.4c	CSS3E04c	Cryptography and Network Security 3.4d
CSS3E04d		Advanced Web Technology
3.4e	CSS3E04e	Virtualisation and Cloud Computing 3.4f
CSS3E04f		Data Warehousing and Data Mining

List of Electives for CSS3E05

No	Course Code	Course Name
3.5a	CSS3E05a	Data Compression
3.5b	CSS3E05b	Pervasive Computing
3.5c	CSS3E05c	System Security
3.5d	CSS3E05d	Molecular Simulation and Modeling 3.5e
CSS3E05e		Fundamentals of Big Data
3.5f	CSS3E05f	Web Engineering

Semester IV

No	Course Code	Course Name	Credit	Weightage		Hrs/wk		
4.1	CSS4E01*	Elective IV	4	100	100	4	1	5
4.2	CSS4C01*	Principles of Software Engineering	2	100	100	2		2
4.2	CSS4C02	Project Work (Duration of the Project = 16 Weeks)	8	25 75	100			
		Total	**14**					
		Total Credits (Sem I – IV)			**87 Credits**			

*Evaluation is to be done Internally for these papers (by providing 25% weightage for continues assessment and 75% weightage for the internal examination)

Note:-

- Evaluation for CSS4C01 and CSS4E01 is to be carried out as follows:
  - 25% weightage for the following components:

Components for Continuous Evaluation	Weightage
Test papers with at least 25% questions based on problems or programs (minimum two)	2
Assignments (minimum two) such as homework, problem solving, group discussions, quiz, literature survey, term- project, software exercises, etc.	1
Regularity in the class	1
Seminar	1
Total	**5**

  - 75% weightage for the End Semester Examination which is to be conducted by the concerned department. Question papers for the examinations are to be prepared in the format specified for university examinations with 36 weightage.
  - Suppose that a student got 3.5 points for the components of continuous evaluation and 3.0 points for the End Semester Examination. The total grade point is to be calculated as follows: (1 X 3.5 + 3 X 3.0)/4 = 3.13.

List of Electives for CSS4E01		
No	**Course Code**	**Course Name**
4.1a	CSS4E01a	Digital Image Processing
4.1b	CSS4E01b	Advanced Topics in Database Design
4.1c	CSS4E01c	Software Development for Portable Devices 4.1d
	CSS4E01d	Storage Area Networks
4.1e	CSS4E01e	Semantic Web
4.1f	CSS4E01f	Advanced Java Programming

Semester I

CSS1C01|Discrete Mathematical Structures

Course Number: 1.1 **L P C**
Contact Hours/Week: 4 **4 0 4**
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To introduce discrete mathematics concepts necessary to understand basic foundation of Computer Science.

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

Reference

1. C Liu and D. Mohapatra, *Elements of Discrete Mathematics - A Computer Oriented Approach*, TMH, ISBN: 1259006395.
2. Alan Doerr and Kenneth Levassur, *Applied Discrete Structure for Computer Science*, Galgotia Publications Pvt. Ltd, ISBN: 9780574217554.
3. J. K. Sharma, *Discrete Mathematics*, Macmillan Publishers India Limited, ISBN: 1403924759.
4. J. P. Tremblay and R. Manohar, *Discrete Mathematical Structures with Application to Computer Science*, McGraw-Hill Companies, ASIN: B001FPXR5Y.

CSS1C02 | Advanced Data Structures

Course Number: 1.2

L	P	C
3	2	4

Contact Hours per Week: 5 (3 Lecture + 2 Practical)
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25 % (Internal) + 75 % (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References:

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, ISBN: 978-0201000238.
2. Horowitz E and Sahni S, *Fundamentals of Data Structures*, Computer Science Press, ISBN: 9780716780427.
3. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, *Fundamentals of Data Structures in C*, Silicon Press, ISBN: 0929306406.
4. Richard F. Gilberg and Behrouz A. Forouzan, *Data Structures: A Pseudocode Approach With C*, Thomson Brooks/Cole Publications, Course Technology, ISBN: 9780534390808.
5. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, *Data Structure using C*, Prentice- Hall, ISBN: 9780131997462.
6. Robert Kruse, Tondo C L and Bruce Leung, *Data Structures & Program Design in C*, Pearson India, 2nd Edition, ISBN: 9788177584233.
7. U. A. Deshpande and O. G. Kakde, *Data Structures & Algorithms*, ISTE Learning Materials Centre, New Delhi, ISBN: 9788188057054.
8. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, *Introduction to Algorithms*, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN: 978-0262033848.
9. Seymour Lipschutz, *Data Structures With C*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070701989.
10. Jean-Paul Tremblay, Paul G. Sorenson, P. G. Sorenson, *Introduction to Data Structures with Applications*, 2nd Edition, Mcgraw-Hill College, ISBN: 0070651574.

CSS1C03 | Theory of Computation

Course Number: 1.3 **Contact Hours per Week:** 4 **Number of Credits:** 4

L	P	C
4	0	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Preliminaries - Introduction to formal proof and inductive proofs - The central concepts of Automata Theory - Alphabets, Strings, Languages – Introduction to automata and grammar - Deterministic Finite Automata, Non-deterministic Finite Automata – Equivalence of Deterministic and Nondeterministic Finite Automata - Finite Automata with Epsilon Transitions - Equivalence of NFA with and without epsilon moves.

Unit II

Regular Expressions, Finite Automata and Regular Expressions, Properties of Regular Languages - Pumping lemma and proof for existence of non regular languages, Closure properties, homomorphism, substitution - Decision Properties - Equivalence and Myhill Nerode and DFA state minimization – Regular Grammar.

Unit III

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

Unit IV

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

Unit V

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

References

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

CSS1C04 | The Art of Programming Methodology

Course Number: 1.4

L	P	C
2	2	4

Contact Hours per Week: 4 (2 Lecture + 2 Practical)
Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure: None.
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Part A: Problem Solving – Flow Chart for Structured Programming – Program Charts – System Charts – Variables, data names, programming statements – Flow Chart Symbols – Terminal Symbols – I/O – Comments – Connectors – Process – Decision - Loops – Flow Charts of Fundamental Algorithms (mentioned in Part B) - **Part B:** Algorithm Design – Problem Solving Aspect – Top Down Design – Formal Conventions – Writing Algorithms – Fundamental Algorithms (Discuss the Design of

Algorithms only). **Part C:** Program, Characteristics of a good program - Modular Approach - Programming style - Documentation and Program Maintenance - Compilers and Interpreters - Running and Debugging Programs - Syntax Errors - Run-Time Errors - Logical Errors - Concept of Structured Programming.

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

1. Martin M. Lipschutz and Seymour Lipschutz, *Schaum's Outline of Theory and Problems of Data Processing,* ISBN: 9780070379831 (Unit I Part A).
2. Anil Bikas Chaudhuri, *The Art Of Programming Through Flowcharts & Algorithms,* Laxmi Publications, New Delhi (Unit I Part A).
3. Jean Paul Trembley and Pual G Sorenson, *An Introduction to Data Structures with Applications,* Tata McGraw Hill (Unit I Part B).
4. R G Dromey, *How to Solve by Computer,* Pearson Education, 5th Edition, ISBN: 0134340019 (Unit I Part B).
5. J.B Dixit, *Computer Fundamentals and Programming in C,* Firewall Media, ISBN: 8170088828. (Unit I Part C).
6. Dennie Van Tassel, *Program Style, Design, Efficiency, Debugging, and Testing,* PHI, ISBN: 0137299478 (Unit I Part C).
7. E Balagruswamy, *Programming in ANSI C*, TMH, 5th Edition, ISBN: 0070681821.
8. Kamthane, Programming in C, 2nd Edition, Pearson India, ISBN: 8131760316.
9. Brian W. Kernighan and Dennis M. Ritchie, C Programming Language, PHI, ISBN: 0131103628.
10. Kanetkar, *Let Us C*, BPB Publications, 8th Edition, ISBN: 1934015253.

CSS1C05 | Computer Organization & Architecture

Course Number: 1.5

L	P	C
4	0	4

Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

Main Memory - memory hierarchy – main memory – RAM, ROM - memory cells - cell organization - working – performance considerations - cache memory – virtual memory - memory management requirements - secondary storage – memory interleaving. Input / Output Organization - Accessing I/O devices – programmed I/O, interrupt I/O - interrupts - interrupt processing – hardware interrupts – programmable interrupt controller – vectored interrupts - interrupt nesting - daisy chaining - direct memory access (DMA) - DMA operations & DMA Controller, Introduction to I/O interfaces, I/O channels, IO Processors.

Unit V

Architecture - General 8-bit microprocessor and its architecture - 8085 - Functional block diagram – architecture functions of different sections - architecture of 8086 CPU. Instruction Sets - Instruction format - addressing modes - instruction set of 8085 CPU - Instruction cycle - timing diagrams - different machine cycles - fetch and execute operations - estimation of execution time - estimation of execution time. Intel 8051 Micro controller – Architecture - basic instructions - basic assembly language programs - peripherals: interrupts, timers, parallel port, serial port.

References

1. V Carl Hamacher, Zvonko Vranesic and Safwat Zaky, *Computer Organization*, Mc-Graw Hill International Edition, 5th Edition, ISBN: 9780071122184.
2. Morris Mano, *Digital Logic and Computer Design*, Prentice Hall of India, ISBN: 0876924178.
3. M Morris Mano, *Computer System Architecture*, Prentice Hall, 3rd Edition. ISBN: 0131755633.
4. William Stallings, *Computer Organization and Architecture*, 9th Edition, Prentice Hall, ISBN: 013293633X.

5. Andrew S Tanenbaum, *Structured Computer Organization*, Prentice Hall, 6th Edition, ISBN: 0132916525.
6. Floyd Thomas L, *Digital Fundamentals*, Pearson Education, 10th Edition, Prentice Hall, ISBN: 0132359235.
7. Albert Paul Malvino, Donald P Leach, *Digital Principles and Applications*, McGraw Hill, 4th Edition, ISBN: 0070398836.
8. Thomas C Bartee, *Digital Computer Fundamentals*, McGraw Hill, 6th Edition, ASIN: B004H0SL5K.
9. Ramesh. S. Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, 6th Edition, Wiley Eastern Ltd, New Delhi, ISBN: 9788187972884.
10. Mohamed Rafiquzzaman, *Introduction to Microprocessors and Microcomputer Based System Design*, 2nd Edition, CRC Press, ISBN: 9780849344756.
11. Muhammad Ali Mazidi, Janice Mazidi, Rolin Mckinlay, Janice M. Mazidi, Janice Gillispie Mazidi and Rolin D., *The 8051 Microcontroller and Embedded Systems*, Pearson Education Asia, 5th Indian Reprint, ISBN: 013119402X.

CSS1P06 | Practical I

Course Number: **1.6**

L	P	C
0	4	4

Contact Hours per Week: **4**
Number of Credits: **4**
Number of Contact Hours: **60 Hrs**
Prerequisite/Exposure: **None**
Course Evaluation: **25% (Internal) + 75% (External)**

Objectives

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

Course Outline

Unit I: C Programming

1. Simple C Programs like area of a circle, checking whether a given number is odd or even.
2. Implementation of programs using Loops (pyramid printing, factorial computation, number reversing, checking for Armstrong numbers, finding first N or Nth Prime numbers etc).
3. Use of 1D and 2D Arrays (searching, sorting and vector operations, matrix addition, matrix multiplication).
4. String Manipulations.
5. Structures and Unions (like addition of two complex numbers, student record creation and manipulation etc).
6. Writing functions.

7. Implementation of recursion (recursive function to compute a factorial, reverse string etc).
8. Command line arguments.
9. Pointers - simple programs to learn concept of pointers, array operation using pointers etc.
10. File operations – file and structures.

Unit 2: Data Structures and Algorithms

1. Implementation of stacks using arrays.
2. Implementation of queues, circular queue using arrays.
3. Implementation of sequential search and binary search techniques.
4. Implementation of linked lists and operations (add, insert, delete, search) on linked lists.
5. Implementation of stacks using linked list.
6. Implementation of queues using linked list.
7. Implementation of doubly linked list.
8. Implementation of circular linked list.
9. Implementation of binary tree and traversals.
10. Implementation of Binary search trees and perform the operations on BST.
11. Implementation of various sorting algorithms.
12. Conversion of an infix expression to the postfix form using stacks. 13.Evaluation of a postfix expression.
14. Implementation of graphs and graph traversals.
15. Implementation of heap tree and operations.

Semester II

CSS2C01 |Design and Analysis of Algorithms

Course Number: 2.1 **L P C**
Contact Hours per Week: 4 (3 Lecture + 1 Practical) **3 1 4**
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Algorithm Design: Introduction, Steps in developing algorithm, Methods of specifying an algorithm, Decisions prior to designing: based on the capabilities of the device, based on the nature of solutions, based on the most suitable data structures. Model of Computation: RAM model and PRAM model. Important Problem Types (Introductory concepts): Sorting, Searching, String processing, Graph problems, Combinatorial problems, Geometric problems and Numerical problems.

Unit II

Basic Technique for Design of Efficient Algorithm: Brute Force approach (String matching), Divide-and-Conquer approach (Merge sort), Branch-and-Bound technique (Knapsack problem). Greedy approach (Kruskal's algorithm and Prim's Algorithm), Dynamic Programming (Longest Common Subsequence), Backtracking (Sum of subsets problem).

Unit III

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

matrix multiplication, Analysis of Merge sort.

Unit IV

Complexity - Complexity Classes: P, NP, NP Hard and NP Complete problems. NP Completeness reductions for Travelling Salesman Problem and Hamiltonian Cycle. P versus NP problem.

Unit V

Analyzing Parallel Algorithms: Time Complexity, Cost, Number of Processors, Space Complexity, Speed up, Efficiency, Scalability, Amdahl's Law. Parallel merging and sorting, Euler tour technique, Parallel prefix computation, Deterministic symmetry breaking.

References:

1. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, *Introduction to Algorithms*, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN: 9780262033848 (Unit I, II, III and IV).
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *The Design and Analysis of Computer Algorithms*, 1st Edition. Addison Wesley, ISBN: 0534915728 (Unit I, II, III and IV).
3. Pallaw, V K, *Design and Analysis of Algorithms*, Asian Books Private Ltd, 2012, ISBN: 8184121687 (Unit I, II, III and IV).
4. Sanjay Razdan, *Fundamentals of Parallel Computing*, Narosa Publishing House, 2014, ISBN: 9788184873481 (Unit V).
5. Pandey H M, *Design and Analysis of Algorithms*, University Science Press, 2013, ISBN: 9788131803349 (Unit I, II, III and IV).
6. Upadhyay N, *Design and Analysis of Algorithms*, SK Kataria & Sons, 2008 (Unit I, II, III and IV).
7. U. Manber, *Introduction to Algorithms: A Creative Approach*, Addison Wesley, ISBN: 9780201003277 (Unit I, II, III and IV).
8. Gilles Brassard and Paul Bratley, *Fundamentals of Algorithmics*, Prentice-Hall of India, ISBN: 0133350681 (Unit I, II, III and IV).
9. Goodman S E and Hedetniemi, *Introduction to the Design and Analysis of Algorithms*, Mcgraw Hill, ISBN: 0070237530 (Unit I, II, III and IV).
10. Horowitz E and Sahni S, *Fundamentals of Computer Algorithms*, Galgotia Publications Pvt. Ltd, ISBN: 8175152575 (Unit I, II, III and IV).
11. Oded Goldreich, *P, NP and NP - Completeness*, Cambridge University Press, 2011. ISBN: 0521122546 (Unit V).

12. Donald Knuth, *The Art of Computer Programming, Fundamental Algorithms*, Volume 1, Addison Wesley, 1997, ISBN: 8177587544 (Unit I).

13. Sanjeev Arora and Boaz Borak, *Computational Complexity - A Modern Approach*, Cambridge University Press; 2009, ISBN: 0521424267 (Unit III).

14. Daniel Hills W and Bruce M Boghosian, *Parallel Scientific Computation*, Science, 13 August 1993, Vol. 261 (5123), pp.856-863 (Unit V).

CSS2C02 |Operating System Concepts

Course Number: 2.2 **L P C**
Contact Hours per Week: 4 (3 Lecture + 1 Practical) 3 1 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: Advanced Data Structures (CSS1C02), Computer Organization & Architecture (CSS1C05)
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

1. William Stallings, *Operating Systems, Internals and Design Principles*, 7th Edition, Pearson, ISBN: 9780273751502.
2. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, *Operating System Concepts*, 9th Edition, John Wiley & Sons, ISBN: 9781118063330.
3. Ann McIver McHoes and Ida M. Flynn, *Understanding Operating Systems*, 6th Edition, Cengage Learning, 2010, ISBN: 9781439079201.
4. Mukesh Singhal and Niranjan G. Shivaratri, *Advanced Concepts in Operating Systems – Distributed, Database, and Multiprocessor Operating Systems*, Tata McGraw-Hill Education Private Limited, ISBN: 9780070575721.
5. Current Literature (for Mobile Operating Systems).

CSS2C03 |Computer Networks

Course Number: 2.3 **Contact Hours per Week:** 4 **Number of Credits:** 4

L	P	C
4	0	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

Transport layer services – introduction – relationship between Transport and Network layer – UDP – reliable data transfer – TCP - congestion control - Network layer services – routing – IP - routing in Internet - router - IPV6 - multicast routing
– mobility.

Unit IV

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

Unit V

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

References

1 J. F. Kurose and K . W. Ross, *Computer Networking: A Top-Down Approach Featuring Internet*, 6th Edition, Perason Education, ISBN: 0132856204.

2 Behrouz Forouzan, *Data Communications and Networking*, 4th Edition, McGraw- Hill Reprint, ISBN: 0073250325.

3 Peterson L.L. and Davie B .S., *Computer Networks, A Systems Approach*, 5th Edition, Morgan Kaufmann, ISBN: 9780123850591.

4 Keshav, *An Engineering Approach to Computer Networking*, Pearson Education Asia, ISBN: 97898123598652000.

5 Andrew S. Tanenbaum, *Computer Networks*, 5th Edition, PHI, ISBN: 9788131787571.

6. Herbert Scheldt, *Java Complete Reference*, 7th Edition, McGraw-Hill Osborne Media, ISBN: 9780072263855.

CSS2C04 |Computational Intelligence

Course Number: 2.4

L	P	C
4	0	4

Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure:
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To introduce concepts of Artificial Intelligence and Machine Learning.

Course Outline

Unit I

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

Unit II

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

Unit III

Knowledge representation issues, representation and mappings, representing simple facts in logic, representing instances and ISA relationships, computable functions and predicates, resolution, natural deduction, knowledge representation using rules,

logic programming, forward versus backward reasoning, symbolic reasoning under uncertainty- nonmonotonic reasoning, depth first search, breadth first search.

Unit IV

Game playing – the Minimax search procedure, adding Alpha-beta cutoffs, additional refinement, iterative deepening, planning system and its components, understanding, understanding as constrained satisfaction. Slot and filler structures: semantic nets, frames, conceptual dependency, scripts. Definition and characteristics of expert system, representing and using domain knowledge, expert system shells. Knowledge engineering, knowledge acquisition, expert system life cycle & expert system tools, CYCIN & DENDRAL examples of expert system.

Unit V

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

References

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

CSS2P06 |Practical II

Course Number: 2.6 **Contact Hours per Week:** 4 **Number of Credits:** 4

L	P	C
0	4	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To practically implement the theory portions covered in *Operating System Concepts (CSS2C02)* and *Computer Networks (CSS2C03).*
- To extend the programming knowledge acquired thru *The Art of Programming Methodology (CSS1C04).*

Course Outline

Unit I: Computer Networks

1. Design a LAN with a given set of requirements. The design should include topology, hardware and software requirements like cable, connectors, hubs/switches/bridges, interface cards along with a budget for the LAN. (Faculty in charge should give the requirements to the students)*.
2. Establish a LAN that consists of at least one server and two clients*.
3. Study of network utilities in Linux/Windows (hostname, ping, ifconfig, ipconfig, netstat, nslookup, telnet, traceroute, finger, telnet, tracert, arp, ftp etc)*.
4. Implementation of TCP Client.
5. Implementation of TCP Server.
6. Write a program to check the Date and Time in TCP Date Time Client.
7. Write a program to check the Date and Time in TCP Date Time Server.
8. Implementation of UDP client and server.
9. Write a program to transfer Files using UDP.
10. Implementation of transferring files using FTP.
11. Write a program to simulate the sliding window protocol.
12. Study of Network Simulators (NS2/Glomosim)*.

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

Unit II: Operating System Concepts

1. Write programs using the following system calls: fork(), execl() and wait().
2. Write File System Calls to write, append and display.
3. To accept the burst time for a set of processes for FCFS scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.
4. To accept the burst time for a set of processes for SJF scheduling and create chart consisting of the burst time, turnaround time and wait time of each process.
5. To accept the burst time and priority for a set of processes for Priority scheduling and create chart consisting of the burst time, priority, turnaround time and wait time of each process.
6. To create n Fibonacci numbers and prepare a list of prime numbers amongst them (use pipe for IPC).
7. To demonstrate IPC using shared memory.
8. To allocate memory requirements for processes using best fit allocation- Accept n processes with their memory requirements and n holes with their sizes. Perform memory allocation using Best Fit algorithm. Display a chart consisting of the process and the allocated hole.
9. To accept n processes with their memory requirements and n holes with their sizes. Perform memory allocation using First Fit algorithm. Display a chart consisting of the process and the allocated hole.
10. To demonstrate the process of contiguous allocation of memory blocks to store files of varying sizes.
11. To implement Producer Consumer problem using semaphores.

CSS2P07 |Term Paper

Course Number: 2.7
Contact Hours per Week: 1
Number of Credits: 1
Number of Contact Hours: 15 Hrs
Prerequisite/Exposure: None Course
Evaluation: 100% (Internal)

L	P	C
0	1	1

Objectives

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

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

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

References

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

Semester II | Elective I CSS2E05 | List of Electives

CSS2E05a | Computer Graphics

Elective I: CSS2E05 Course Number: 2.5a Contact Hours per Week: 4

L	P	C
4	0	4

Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

1. Donald Hearn and M. Pauline Baker, *Computer Graphics*, 2nd Edition,Prentice Hall, ISBN: 0135309247.
2. Donald D. Hearn, M. Pauline Baker and Warren Carithers, *ComputerGraphics with Open GL*, 4th Edition, Prentice Hall, ISBN: 9780136053583.
3. Hill, *Computer Graphics using OpenG,L*, 3rd Edition, Prentice Hall of India Private Ltd. New Delhi, ISBN: 8120338294.
4. Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, Dave Shriner and Tom David, *Open GL Programming Guide*, 6th Edition, Person, ISBN: 9780201604580.
5. The Official Guide to Learning OpenGL, Version 1.1, Available at http://www.glprogramming.com/red/.
6. Shreiner and Angel, *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*, 6th Edition, Pearson Education, ISBN: 0132545233.

CSS2E05b | Introduction to Soft Computing

Elective I: CSS2E05 Course Number: 2.5b Contact Hours per Week: 4 Number of Credits: 4

L	P	C
4	0	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

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

CSS2E05c | Web Technology

Elective I: CSS2E05

L	P	C
4	0	4

Course Number: 2.5c
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To introduce the tools for creating and maintaining websites – content development (HTML), client side scripting (JavaScript), web server (Apache), server side scripting (PHP) and content management system (Joomla!).

Course Outline

Unit I

Introduction to web programming – introduction to SGML features – HTML, XHTML, DHTML, XML – HTML Vs XML – creating XML documents – parsing an XML document – writing well formed documents – organizing elements with namespaces – defining elements in a DTD – declaring elements and attributes in a DTD. Overview of HTML

- basic formatting tags - heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image. Attributes - align, color, bgcolor, font face, border, size. Navigation links using anchor tag - internal, external, mail and image links. Lists - ordered, unordered and definition, table tag, HTML form controls - form, text, password, textarea, button, checkbox, radio button, select box, hidden controls, frameset and frames. CSS.

Unit II

Client side programming – introduction – popular client side scripting languages - Java Script - introduction, identifiers, operators, functions, event handling, classes, objects, array, math, string, window object, navigator DHTML font, text, image change, table expansion. JavaScript's object model- strengths and weaknesses of JavaScript - building and extending objects in JavaScript - events in JavaScript - event handlers - creating interactive forms – cookies - storing users choices in cookies - encoding cookies - browser objects - object hierarchy, creating browser objects, working with window, document, history & location - browser detection, Java to JavaScript communication.

Unit III

Web server – role - Apache web server – introduction – architecture – features - Apache's role in the Internet – LAMP – WAMP - installation and configuration - build and install Apache web server - verify initial configuration start, stop, and status the Apache server process. Configure Apache core modules security - basic security with Apache - host-based authentication - user-based authentication - secure sockets layer (SSL) - delivering dynamic web content - Apache's role in the dynamic web - server side includes (SSIs) - configure Apache web server to support CGI – CGI Alternative Technologies. virtual hosts, redirection, indexing – virtual hosting with Apache, virtual host configuration redirection, directory indexing. Proxy servers and firewalls - apache proxy configuring, proxy services firewalls and apache, firewall architecture models monitoring apache web server - error logs, logging http access, web server status and server information, user tracking - proxy caching.

Unit IV

Server side programming – server side scripts – PHP – designing dynamic web pages using PHP - defining PHP variables – variable types – operators – control flow constructs in PHP – passing form data between pages - establishing connection with MySQL database – managing database.

Unit V

Overview of content management system - coding for reusability (header.php) – user management - article publishing - additional CMS features – Web site development using Joomla!.

References

1. Thomas A. Powell, *The Complete Reference HTML*, 3rd Edition, McGraw-Hill/Osborne Media, ISBN: 0072129514.
2. Thomas A. Powell, *Web Design: The Complete Reference*, 2nd Sub-Edition, McGraw-Hill/Osborne Media, ISBN: 0072119772
3. Robert W. Sebesta, *Programming with World Wide Web*, 7th Edition,Addison-Wesley, ISBN: 9780132665810.

4. Xue Bai, Michael Ekedahl, Joyce Farrell, Don Gosselin, Diane Zak, Bill Morrissey, Michael V. Ekedahl, Peter Macintyre and Shashi Kaparthi, *The Web Warrior Guide to Web programming*, Thomson Learning, ISBN: 9780619064587.
5. Chris Bates, *Web Programming: Building Internet Applications*, 3rd Edition, Wiley Academic Catalog, ISBN: 9780470017753.
6. Paul J. Deitel, Harvey M. Deitel, Harvey Deitel, Paul Deitel and Abbey Deitel, *Internet and World Wide Web: How to Program*, 5th Edition, Prentice Hall, ISBN: 9780132151009.
7. R. Allen Wyke and Richard Wagner, *JavaScript Unleashed*, 3rd Edition, SAMS, ISBN: 9780672317637.
8. Richard Bowen Ken Coar, Ken A Coar and Matthew Marlowe, *Apache Server Unleashed*, SAMS, ISBN:0672318083.
9. Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K Glass, *Beginning PHP5, Apache, and MySQL Web Development*, Wrox, ISBN: 0764579665.
10. Jennifer Marriott and Elin Waring, *The Official Joomla! Book*, Addison-Wesley Professional, ISBN: 978-0321821546.
11. Ron Severdia and Kenneth Crowder, *Using Joomla: Building Powerful and Efficient Web Sites*, 1st Edition, O'Reilly Media, ISBN: 9780596804947.

CSS2E05d | Bioinformatics

Elective I: CSS2E05

L	P	C
4	0	4

Course Number: 2.5d
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- Expose students to the popular genomic and proteomic databases and to impart knowledge in processing and analyzing genomic data.
- Introduce advanced topics in Bioinformatics.

Course Outline

Unit I

Bioinformatics - introduction to - nature and scope of computational biology and Bioinformatics. Cells - prokaryotes and eukaryotes - DNA double helix - central dogma – RNA, Amino acids, Proteins - string representations. A glossary of Bioinformatics terms - file format for bio-molecular sequences, sequence alignment, phylogeny, gene finding, microarray analysis, homology and evolutionary relationships.

Unit II

Basic algorithms in Computational Biology - exhaustive search methods and their applications in Computational Biology - string matching algorithms. Motif finding - tandem repeats – concept of dynamic programming - graph algorithms - clustering algorithms.

Unit III

Sequence alignment - pair-wise sequence alignment, need of scoring schemes - penalizing gaps, scoring matrices for amino acid sequence alignment, PAM probability matrix and log odds matrix, BLOSUM, Dot-plot visualization, Needleman- Wunsch algorithm- effect of scoring schemes – evalues - BLAST and FASTA, Smith – Waterman algorithm for local alignment.

Unit IV

Multiple sequence alignment - sequence alignment using dynamic programming, N-dimensional dynamic programming. Tools for MSA - muscle and T-Coffee. Phylogenetic algorithms - evaluation of phylogenetic trees, significance.

Unit V

Introduction to the major resources - NCBI, EBI and ExPASy - nucleic acid sequence databases - GenBank, EMBL, DDBJ – Protein sequence databases - SWISS-PROT, TrEMBL, PIR_PSD - genome databases at NCBI, EBI, TIGR, SANGER – procedures to access these databases and to make use of the tools available.

Text Books

1. Mount D, Bioinformatics*: Sequence & Genome Analysis*, 2nd Edition, Cold spring Harbor Press, ISBN: 978-087969712.
2. Dan Gusfield, *Algorithms on Strings Trees and Sequences*, 1st Edition, Cambridge University Press, ISBN: 0521585198.
3. Pevzner P A, *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, Cambridge, MA, ISBN: ISBN: 9780262161978.
4. Jeremy J. Ramsden, *Bioinformatics: An Introduction*, Springer, ISBN: 9789401570961.
5. Sushmita M and Tinku A, *Data Mining: Multimedia, Soft Computing and Bioinformatics*, Wiley-Interscience, ISBN: 9780471460541.

References

1. Richard M. Karp, *Mathematical Challenges from Genomics and Molecular Biology*, Notices of the American Mathematical Society, vol. 49, no. 5, pp. 544-553.
2. Glyn Moody, *Digital Code of Life: How Bioinformatics is Revolutionizing Science, Medicine and Business*, ISBN: 9780471327882.
3. Tao Jiang, Ying Xu and Michael Q. Zhang, *Current Topics in Computational*

Molecular Biology Edible Oil Processing, 1st Edition, AneBooks Pvt Ltd, ISBN: 9788180520525.

4 Andrzej K. Konopka and M. James C. Crabbe, *Compact Handbook of Computational Biology*, 1st Edition, CRC Press, ISBN: 9780824709822.

5 Richard E. Bellman, *Dynamic Programming*, Princeton University Press, ISBN: 9780691146683.

6 Needleman S B and Wunsch C D, *A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins*, J. Mol. Biol., 48 (1970) 443–453.

7. Smith T F and Waterman M S, *Identification of Common Molecular Subsequences*, J. Mol. Bio. 147 (1981) 195–197.

8 Watson J D and Crick F H C, *A Structure for Deoxyribose Nucleic Acid*, Nature, 171 (1953) 737–738.

9 Pevzner P A and Waterman M S, *Open Combinatorial Problems in Computational Molecular Biology*, Proc. Third Israel Symp. Theo. Comp. Syst. IEEE Computer Society Press, (1995) 158 – 173.

CSS2E05e |Computer Optimization Techniques

Elective I: CSS2E05 Course Number: 2.5e Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

L	P	C
4	0	4

Objectives

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

Course Outline

Unit I

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

Unit II

Transportation and Network models - The transportation models and algorithm, the assignment and transshipment models, minimum spanning tree algorithm,

shortest-route problem, maximum flow and min-cost models, critical path method and algorithms for matching.

Unit III

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

Unit IV

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

Unit V

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

References

1. H. A. Taha, *Operations Research: An Introduction*, 9th Edition, Pearson Prentice Hall, ISBN: 013255593X.
2. C. H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications,ISBN: 9780486402581.

CSS2E05f |Numerical and Statistical Methods

Elective I: CSS2E05

	L	P	C
Course Number: 2.5f	**4**	**0**	**4**

Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Approximation and errors in computing - introduction, significant digits - inherent errors – numerical error - modeling errors - blunders - absolute and relative errors - conditioning and stability. Roots of non-linear equations - introduction - iterative methods – bisection - false position – Newton - Raphson's, Secant and Bairstow's methods.

Unit II

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

Unit III

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

Unit IV

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

Unit V

Introduction linear programming – mathematical formulation – graphical method of solution – simplex method – duality – dual simplex – transportation – assignment problems.

References

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

Semester III

CSS3C01 | Advanced Database ManagementSystem

Course Number: 3.1 **L P C**
Contact Hours per Week: 5 **4 1 4**
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

Relational database query languages - basics of QBE and SQL. Data definition in SQL - data types, creation, insertion, viewing, updation, deletion of tables, modifying the structure of the tables, renaming, dropping of tables. Data constraints – I/O constraints, primary key, foreign key, unique key constraints, ALTER TABLE command - database manipulation in SQL - computations done on table data - SELECT command, logical operators, range searching, pattern matching, grouping data from tables in SQL, GROUP BY, HAVING clauses. Joins – joining multiple tables, joining a table to it. DELETE – UPDATE. Views - creation, renaming the column of a view, destroys view. Program with SQL - data types Using SET and SELECT commands, procedural flow, IF, IF /ELSE, WHILE, GOTO, global variables. Security - locks, types of locks, levels of locks. Cursors - working with cursors, error handling, developing stored procedures, CREATE, ALTER and DROP, passing and returning data to stored procedures, using stored procedures within queries, building user defined functions, creating and calling a scalar function, implementing triggers, creating triggers, multiple trigger interaction (Use MySQL as the RDBMS).

Unit IV

Transaction management, concurrency control and query processing - concept, definition and states of transactions, ACID properties – concurrency control, serializability – conflict serializability, view serializability, recoverability-recoverable schedules, non-cascading schedules, strict schedules. concurrency control schemes - locking- two phase locking, deadlock, granularity, timestamp ordering protocol. Basics of query processing.

Unit V

Object Oriented Database Management Systems (OODBMS) - concepts, need for OODBMS, composite objects, issues in OODBMSs, advantages and disadvantages of OODBMS. Distributed databases - motivation - distributed database concepts, types of distribution, architecture of distributed databases, the design of distributed databases, distributed transactions, commit protocols for distributed databases.

Reference

1. Elmasri and Navathe, *Fundamentals of Database Systems*, 5th Edition, Pearson, ISBN: 9788131758984.
2. Abraham Silbersehatz, Henry F. Korth and S.Sudarshan, *Database System Concepts*, 6th Edition, Tata McGraw-Hill, ISBN: 0071325220.
3. CJ Date, *An Introduction to Database Systems*, 8th Edition, Addison Wesley, ISBN: 0321197844.
4. Ramakrishnan and Gehrke, *Database Management Systems*, 3rd Edition, McGraw - Hill Education, ISBN: 9339213114.
5. Alexis Leon and Mathews Leon, *Database Management Systems*, 1st Edition, Vikas Publishers, ISBN: 8182092221.
6. Vikram Vaswani, *MySQL The complete Reference*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070586845.

7. Joel Murach, *Murach's Mysql*, Mike Murach & Associates Inc, ISBN: 9350237695.
8. Paul DuBois, *MySQL Cookbook*, 2nd Edition, O'Reilly Media, ISBN: 8184042809.

CSS3C02 | Principles of Compilers

Course Number: 3.2 **L P C**
Contact Hours per Week: 4 **4 0 4**
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: **Advanced Data Structures (CSS1C02), Operating System Concepts (CSS2C02)**
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

Syntax analysis – role of parser – error handling and recovery – definitions of parsing, top-down parsing and bottom-up parsing - context free grammars – derivations - parse tree – ambiguity – associativity and precedence of operators - writing a grammar – top-down parsing – recursive descent parsing - FIRST and FOLLOW – LL (1) Grammars – recursive predictive parsing - bottom up parsing – reductions – handle pruning – shift reduce parsing - operator precedence parsing, simple LR parsing.

Unit III

Intermediate code generation – DAG – three address code – addresses and instructions – quadruples – triples – Static Simple Assignment form – types and declarations – type expressions - type equivalences – declarations – type checking – rules – type conversion – function and operator overloading – type inference and polymorphic functions – control flow – boolean expressions – short circuit code –

flow-control statements – control-flow translation for boolean expressions – BREAK CONTINUE and GOTO statements.

Unit IV

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

Unit V

Code generation – issues in the design of a code generator – the target language – a simple target machine model – the program and instruction costs – address in the target code – static allocation – stack allocation – run-time address for names – basic blocks and flow graphs – representation of flow graphs. Code optimization - the principal sources of optimization – data flow analysis – abstraction – data flow analysis schema – data flow schemas on basic blocks – reaching definitions – live variable analysis – available expressions. Region based analysis – regions – region hierarchies for reducible flow graphs – overview of a region based analysis.

References

1. V Aho A, Ravi Sethi, D Ullman J, *CompilersPrinciples, Techniques and Tools*, 2nd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131721019.
2. K. V. N. Sunitha, *Compiler Construction*, Pearson, ISBN:9789332500297.
3. W Appel and Andrew, *Modern Compiler Implementation in C*, 1st Edition, Cambridge University Press, ISBN: 817596071X.
4. Allen I Holub, *Compiler Design in C*, 1st Edition, PHI Learning Pvt Ltd, ISBN: 812030778X.
5. Tremblay and Sorenson, *The Theory and Practice of Compiler Writing*, 1st Edition, BSP Books Pvt Ltd, ISBN: 8178000776.
6. Torben Ægidius Mogensen, *Basics of Compiler Design*, Department of Computer Science, University of Copenhagen (Online Edition).

CSS3C03 | Object Oriented Programming Concepts

Course Number: 3.3 **Contact Hours per Week:** 4 **Number of Credits:** 4

L	P	C
4	0	4

Number of Contact Hours: 60 Hrs

Prerequisite/Exposure: The Art of Programming Methodology (CSS1C04)

Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Introduction to OOPS - basic principles of object orientation (objects , attributes and methods, encapsulation and information hiding, state retention, object identity, messages, class hierarchy, inheritance, polymorphism, genericity) - introduction to Java - history, versioning, the Java Virtual Machine, byte code, features of Java, language components - primitive data types, comments, keywords, literals, variables scope & declarations, control structures - FOR, IF, WHILE, DO WHILE, SWITCH, BREAK, CONTINUE statements - operators - casts and conversions - arrays.

Unit II

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

Unit III

Exceptions, threads & IO in Java - The file and standard streams, stream classes and interfaces, using byte streams and character streams, threads - threads vs. processes, creating threads, runnable interface, thread class, inter thread communication, synchronization. Exceptions - basic of Java exception handling, hierarchy, developing user defined exception classes.

Unit IV

Applets, AWT & Swing - applet class, types of applet, skeleton, applet tag, passing parameters, event handling, delegation event model, event classes, listeners, AWT

classes and window fundamentals, frames, working with fonts, graphics and colors, AWT controls, layouts and menus, dialogue boxes. Swings - Japplets, icon, labels, buttons, textbox, combo box, tables and panes.

Unit V

Database and sockets – JDBC - introduction, architecture, drivers, connections, statements, resultset and meta data (Use MySQL as the RDBMS). Sockets: introduction to networking, InetAddress, url, socket, server sockets, datagrams.

Introduction to Unified Modelling Language (UML), UML diagrams, class diagrams, object interaction diagrams, state and activity diagrams, component diagrams, deployment diagrams. Introduction to analysis - object oriented system analysis, design and implementations.

References

1. Herbert Scheldt, *Java Complete Reference*, 8th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 1259002462.
2. E Balaguruswamy, *Programming in Java, : A Primer*, 4th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 007014169X.
3. Kathy Sierra, *Head First Java*, 2nd Edition, Shroff Publishers and Distributors Pvt Ltd, ISBN: 8173666024.
4. David Flanagan, Jim Farley, William Crawford and Kris Magnusson, *Java Enterprise in a Nutshell: A Desktop Quick Reference*, 3rd Edition, O'Reilly Media, ISBN: 0596101422.
5. Grady Booch, James Rumbaugh and Ivar Jacobson, *The Unified Modeling Language User Guide*, 2nd Edition, Pearson, ISBN: 8131715825.

CSS3P06 | Practical III

Course Number: 3.6

L	P	C
0	4	4

Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To practically implement the theoretical aspects covered in *Advanced Database Management System (CSS3C01)* and *Object Oriented Programming Concepts (CSS3C03).*
- To extend the programming knowledge acquired through *The Art of Programming Methodology (CSS1C04)* to encompass object oriented techniques.

Course Outline

Unit I: Advanced Database Management System

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

Unit II: Object Oriented Programming Concepts

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

Semester III | Elective II CSS3E04 | List of Electives

CSS3E04a | Pattern Recognition

Elective II: CSS3E04 Course Number: 3.4a Contact Hours per Week: 4

L	P	C
4	0	4

Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

1. Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern Classification*, CBS Publishers & Distributors, 2nd Edition, ISBN: 9788126511167.
2. Gonzalez R.C. and Thomson M.G., *Syntactic Pattern Recognition: An Introduction*, 1st Edition, Addison-Wesley, ISBN: 0201029316.
3. Fu K. S., *Syntactic Pattern Recognition and Applications*, Prentice Hall, ISBN: 0138801207.
4. Rajjan Shinghal, *Pattern Recognition: Techniques and Applications*, 1st Edition, Oxford University Press India, ISBN: 0195676858.

CSS3E04b | Wireless & Mobile Networks

Elective II: CSS3E04

L	P	C
4	0	4

Course Number: 3.4b
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: Computer Networks (CSS2C03), Object Oriented Programming Concepts (CSS3C03) Course
Evaluation: 25% (Internal) + 75% (External)

Objectives

- To understand the fundamental concepts of wireless and mobile networks.
- To familiarize with wireless application Protocols to develop mobile content applications.
- To understand about the security aspects of wireless networks.
- To learn programming in the wireless mobile environment.

Course Outline

Unit I

Introduction - applications - brief history of wireless communication – open research problems – wireless transmission – frequencies for radio transmission – signals – antennas – signal propagation – multiplexing – modulation – spread spectrum – cellular systems – medium access control – motivation – SDMA – FDMA – TDMA – CDMA – comparison.

Unit II

Different generations of Wireless Cellular Networks - 1G, 2G, 2.5G, 3G, 4G. Telecommunication systems – GSM – DECT – TETRA – UMTS – IMT-2000. Wireless LAN – Infrared Vs Radio transmission – Infrastructure Vs Adhoc networks – IEEE 802.11 – HIPERLAN – Bluetooth.

Unit III

Mobile network layer - Mobile IP – Dynamic Host Configuration Protocol - Routing – DSDV – DSR – Alternative Metrics. Transport and application layers - traditional TCP – classical TCP improvements – WAP, WAP 2.0.

Unit IV

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

Unit V

Java for wireless devices - setting up the development environment - basic data types, libraries (CLDC, MIDP) - UI controls - displayable and display image - events and event handling - list and choice - text box - alerts - persistent storage - record stores – records - record enumeration - network MIDlets - the connection framework
- connection interface - connection using HTTP - datagram connection.

References

1. Jochen Schiller, *Mobile Communications*, Pearson Education, 2nd Edition, ISBN: 8131724263.
2. Raj Kamal, *Mobile Computing*, 2nd Edition Oxford Univ Press, ISBN: 0198068913.
3. William Stallings, *Network Security Essentials Applications and Standards*, 4th Edition, Pearson India, ISBN: 8131761754.
4. Yu Feng and Jun Zhu, *Wireless Java Programming with J2ME*, 1st Edition, Sams, ISBN: 0672321351.
5. Dreamtech Software Team, *Wireless Programming with J2ME: Cracking the Code*, Wiley, ISBN: 0764548859.
6. William Stallings, *Wireless Communications and Networks*, 2nd Edition, Pearson India, ISBN: 8131720934.
7. Jochen Burkhardt, Horst Henn, Stefan Hepper, Klaus Rindtorff and Thomas Schaeck, *Pervasive Computing Technology and Architecture of Mobile Internet Applications*, 14th Edition, Pearson Education, ISBN: 8177582801.
8. Nishit Narang and Sumit Kasera, *2G Mobile Networks: GSM and HSCSD*, Tata McGraw Hill Education, ISBN: 0070621063.
9. Hasan Ahmed, Roopa Yavagal and Asoke K Talukder, *Mobile Computing: Technology, Applications and Service Creation*, 2nd Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070144575.

CSS3E04c | Cryptography and Network Security

Elective II: CSS3E04 Course Number: 3.4c Contact Hours per Week: 4 Number of Credits: 4

L	P	C
4	0	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure:
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To be familiar with classical and modern encryption and decryption techniques and apply in the security system.
- To understand various aspects of network security standards.

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

1. William Stallings, *Network Security Essentials Applications and Standards*, 4th Edition, Pearson India, ISBN: 8131761754.
2. William Stallings, *Cryptography and Network Security : Principles and Practice*, 6th Edition, Pearson India, ISBN: 9332518777.
3. Atul Kahate, *Cryptography and Network Security*, 3rd Edition, Tata McGraw- Hill Publishing, ISBN: 9789332900929.
4. Eric Maiwald, *Fundamental of Network Security*, 1st Edition, Tata McGraw - Hill Education, 0071070931.
5. Charlie Kaufman, Radia Perlman and Mike Speciner, *Network Security: Private Communication in Public World*, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120322134.

CSS3E04d| Advanced Web Technology

Elective II: CSS3E04

L	P	C
4	0	4

Course Number: 3.4d
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To introduce the advanced concepts of web development tools – Web 2.0, Web Services, Python, SQLite and MVC architecture.

Course Outline

Unit I

Web 2.0 - definition, characteristics, key features, client side technologies (Ajax and JavaScript frameworks - YUI library, Dojo toolkit, MooTools, jQuery, Ext JS and prototype JavaScript framework), server side technologies (Ruby, Perl, Python, Enterprise Java J2EE and Microsoft.NET Framework), concepts (Rich Internet Application — Web-Oriented Architecture — Social Web), SLATES.

Unit II

Fundamentals of Web Services - Definition, Components, benefits, behavioral characteristics. Web services architecture - web service roles, web service protocol stack, service transport. web services components - XML-RPC, SOAP, WSDL, UDDI. web services security (notions) - confidentiality (XML-RPC and SOAP run on top of HTTP) - support for Secure Sockets Layer (SSL) for HTTP - encrypted communication via SSL, authentication (HTTP's built-in support for Basic and Digest authentication - SOAP security extensions - Digital Signature – SOAP - DSIG -SAML).

Unit III

Introduction to Python – installation – Python interpreter – usage and customization – editor setup – variables, expressions and statements – functions. Strings – lists – list comprehensions – stacks – queues – tuples – sequences – sets – dictionaries – sets - modules, I/O and exception handling - modules – search path – compiled modules – standard modules – packages – input and output functions – files – read and write – exception – handling and raising – user defined exceptions.

Unit IV

Server side programming using Python - server side scripting - CGI - role of Web server – Apache web server – Python server side script – developing Python Server Side Pages (PSP) – capturing form data – validation – processing data – exchange of data between form and server.

Unit V

Python-SQLite integration - features of SQLite, data types, introduction to SQL commands - SELECT, DELETE, UPDATE, INSERT. Python functions for SQLite operations – database connection, database and table creation, selection, query, fetching results - insertion and deletion of data using Python - displaying data from SQLite in webpage. Case study - server MVC design pattern – Django.

References

1. James Governor, *Web 2.0 Architectures : What Entrepreneurs & Information Architects Need to Know*, 1st Edition, Shroff Publisher & Distributors, ISBN: 8184047355.
2. S. V. Subrahmanya and B. V. Kumar, *Web Services: An Introduction*, 2nd Edition, Tata Mc-graw Hill Publishing Co. Ltd, ISBN: 1259002764.
3. Web 2.0, http://en.wikipedia.org/wiki/Web_2.0
4. Web Services, http://www.tutorialspoint.com/webservices/
5. Ron Schmelzer, Michael Qualls, Sam Hunting, David Houlding, Madhu Siddalingaiah, Jason Bloomberg, Travis Vandersypen, Chad Darby and Diane Kennedy, *XML and Web Services Unleashed*, Sams, ISBN:0672323419.

6. Sandeep Chatterjee, James Webber, *Developing Enterprise Web Services: An Architect's Guide*, 1st Edition, Pearson India, ISBN: 8131713172.
7. The Python Tutorial, http://docs.python.org/3.3/tutorial/
8. Allen Downey, Jeffrey Elkner and Chris Meyers, *How to Think Like a Computer Scientist: Learning with Python*, Createspace, *2009*, ISBN: 1441419071. Online Version: http://openbookproject.net/thinkcs/python/english3e/
9. Python Documentation. Available at http://www.python.org/doc/
10. Swarooph CH, *A Byte of Python*. Available at http://swaroopch.com/notes/python/
11. Wesley J Chun, *Core Python Programming*, 2nd Edition, Pearson Education, ISBN: 8131711889.

CSS3E04e| Virtualisation and Cloud Computing

Elective II: CSS3E04 **L P C**
Course Number: 3.4e **4 0 4**
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Introduction - evolution of cloud computing – system models for distributed and cloud computing – NIST cloud computing reference architecture – Infrastructure as a Service (IaaS) – resource virtualization – Platform as a Service (PaaS) – cloud platform & management – Software as a Service (SaaS) – available service providers.

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

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

7. Open stack Operations Guide, http://docs.openstack.org/ops/.
8. Tom White, *Hadoop: The Definitive Guide*, O'Reilly Media, ISBN: 9780596551360.

CSS3E04f | Data Warehousing and Data Mining

Elective II: CSS3E04 Course Number: 3.4f Contact Hours per Week: 4 Number of Credits: 4

L	P	C
4	0	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Data warehouse – definition – operational database systems Vs data warehouses – multidimensional model – from tables and spreadsheets to Data Cubes – schemas for multidimensional databases – measures – concept hierarchies - OLAP operations in the multidimensional data model – data warehouse architecture.

Unit II

Data mining – introduction – definition - data mining functionalities – major issues in data mining - data preprocessing – data cleaning – data integration and transformation – data reduction – data discretization and concept hierarchy generation. Association rule mining - efficient and scalable frequent item set mining methods – mining various kinds of association rules – association mining to correlation analysis – constraint-based association mining.

Unit III

Classification and prediction - issues regarding classification and prediction – classification by decision tree introduction – Bayesian classification – rule based classification – classification by back propagation – support vector machines – associative classification – lazy learners – other classification methods – prediction – accuracy and error measures – evaluating the accuracy of a classifier or predictor – ensemble methods – model section.

Unit IV

Cluster analysis - types of data in cluster analysis – a categorization of major clustering methods – partitioning methods – hierarchical methods – density-based methods – grid-based methods – model-based clustering methods – clustering high dimensional data – constraint-based cluster analysis – outlier analysis.

Unit V

Graph mining - mining object, spatial, multimedia, text and web data - multidimensional analysis and descriptive mining of complex data objects – spatial data mining – multimedia data mining – text mining – mining the World Wide Web.

References

1. Jain Pei, Jiawei Han and Micheline Kamber, *Data Mining Concepts and Techniques*, 3rd Edition, Elsevier, ISBN: 9380931913.
2. Alex Berson and Stephen J. Smith, *Data Warehousing, Data Mining & OLAP*, Computing Mcgraw-Hill, ISBN: 0070062722.
3. K.P. Soman, Shyam Diwakar and V. Ajay, *Insight into Data mining Theory and Practice*, 1st Edition, Prentice Hall of India, ISBN: 8120328973.
4. G. K. Gupta, *Introduction to Data Mining with Case Studies*, 3rd Edition, PHI Learning Pvt. Ltd, ISBN: 8120350022.
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, *Introduction to Data Mining*, 1st Edition, Pearson India, ISBN: 9332518653.

Semester III | Elective III CSS3E05 | List of Electives

CSS3E05a | Data Compression

Elective III: CSS3E05 Course Number: 3.5a Contact Hours per Week: 4

L	P	C
4	0	4

Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Introduction to database systems, file systems Vs DBMS, view of data – data abstraction, view levels, data models, instances and schemas, data independence, database languages, database architecture, database users, database administrator, role of DBA. The entity – relationship (ER) model - entity sets, relationship sets, attributes, constraints, mapping cardinalities, keys, ER diagrams, weak entity sets, strong entity sets.

Unit II

Dictionary methods - string compression, LZ77 sliding window, MZW, GIF images. Image compression - approaches to image compression, intuitive methods and image transform, test images, JPEG, progressive image compression, vector quantization.

Unit III

Wavelet methods - Fourier transform, frequency domain, Fourier image compression, CWT and inverse CWT, Haar transform, filter bank, DWT, JPEG 2000. Video compression - analog video, composite and component video, digital video, video compression, MPEG.

Unit IV

Audio compression - sound, digital audio, human auditory system, MPEG-1 audio layer. Fractal based compression - IFS. Comparison of compression algorithms. Implementation of compression algorithms.

References

1. David Solomon, *Data Compression: The Complete Reference*, 4th Edition, Springer, ISBN: 8184898002.
2. Stephen Welstead, *Fractal and Wavelet Image Compression Techniques*, Lap Lambert Academic Publishing, ISBN: 384651845X.
3. Khalid Sayood, *Introduction to Data compression*, 4th Edition, Elsevier India Pvt. Ltd, ISBN: 8131234088.

CSS3E05b | Pervasive Computing

Elective III: CSS3E05

L	P	C
4	0	4

Course Number: 3.5b
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Introduction to pervasive computing - past, present, future - the pervasive computing market, m-Business, challenges and future of pervasive computing. Application examples of pervasive computing: retail, airline check-in and booking, sales force automation, healthcare, tracking, car information systems, Email access via WAP and voice.

Unit II

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

Unit III

Web application concepts for pervasive computing - history, WWW architecture, protocols, trans-coding, client authentication via the Internet for pervasive computing. WAP and beyond - introduction, components of the WAP architecture, WAP infrastructure, WAP security issues, Wireless Markup Language, WAP push, products, i-Mode, outlook.

Unit IV

Web voice technology - basics of speech recognition, voice standards, speech applications, speech and pervasive computing, security personal digital assistants - history, device categories, personal digital assistant operating systems, device characteristics, software components, standards, mobile applications and personal digital assistant browsers. Server side programming (Java) for pervasive computing - Java 2 Enterprise Edition (Overview), servlets, Enterprise Java Beans, Java Server Pages, Extensible Markup Language, Web Services, Model-View-Controller pattern.

Unit V

Pervasive web application architecture - background, scalability & availability - development of pervasive computing web applications, pervasive application architecture - example pervasive application - introduction, user interface overview, architecture, implementation. Access from PCs - smart-card authentication via the Internet, ordering goods. Access via WAP - WAP functionality, implementation - access from personal digital assistants - extending the example application to personal digital assistants, implementation for synchronized devices, implementation for intermittently connected devices, implementation for connected devices - access via voice: extending the example application to voice access, implementation.

References

1. Jochen Burkhardt, Horst Henn, Stefan Hepper, Thomas Schaec and Klaus Rindtorff, *Pervasive Computing: Technology and Architecture of Mobile Internet Applications*, 14th Edition, Pearson Education, ISBN: 8177582801.
2. Stefen Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*, Wiley India Pvt Ltd, ISBN:8126527331.
3. Guruduth S. Banavar, Norman H. Cohen and Chandra Narayanaswami, *Pervasive Computing: An Application-Based Approach*, Wiley-Blackwell, ISBN: 0471777404.
4. Frank Adelstein, S K S Gupta, GG Richard and L Schwiebert, *Fundamentals of Mobile and Pervasive Computing*, Tata McGraw-Hill, New Delhi, ISBN: 0070603642.
5. Genco and S. Sorce, *Pervasive Systems and Ubiquitous Computing*, 1st Edition, WIT Press, ISBN: 1845644824.
6. Somprakash Bandyopadhyay, Amitava Mukherjee and Debashis Saha, *Networking Infrastructure for Pervasive Computing Enabling Technologies and Systems*, 1st Edition, ISBN: 8184898037.

CSS3E05c | System Security

Elective III: CSS3E05 Course Number: 3.5c Contact Hours per Week: 4

L	P	C
4	0	4

Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure:
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To provide an understanding of the differences between various forms of computer security, where they arise, and appropriate tools to achieve them.

Course Outline

Unit I

Notion of different types of securities - information security - computer security - security goals, relation between security, confidentiality, integrity, availability and authorization, vulnerabilities - principles of adequate protection. Notions of operating security, database security, program security, network security. attacks - threats, vulnerabilities and controls. The kind of problems - interception, interruption, modification, fabrication. Computer criminals - amateurs, crackers, career criminals. Methods of defence - control, hardware controls, software controls, effectiveness of controls.

Unit II

Program security - secure programs - fixing faults, unexpected behaviour, types of flaws. Non-malicious program errors - buffer overflows, incomplete mediation. Viruses and other malicious code - kinds of malicious code, how viruses attach, how viruses gain control, prevention, control example - the brain virus, the internet worm, web bugs. Targeted malicious code - trapdoors, Salami attack. Controls against program threats - development controls, peer reviews, hazard analysis.

Unit III

Operating system security - protected objects and methods of protection - memory address protection - fence, relocation, base/bounds registers, tagged architecture, segmentation, paging. Control of access to general objects - directory, access control list. File protection mechanism – basics forms of protection, single permissions. Authentication - authentication basics, password, authentication process challenge - response, biometrics. Trusted operating systems - security policies for operating systems, models of security - requirement of security systems, multilevel security, access security, limitations of security systems. Trusted operating system design - elements, security features, assurance, system flaws and assurance methods.

Unit IV

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

Unit V

Administrating security - security planning - contents of a security planning, team members, commitment to a security plan, business continuity plans. Risk analysis - the nature of risk, steps of risk analysis. Arguments for and against risk analysis, organizational security policies - purpose and goals of organizational security. Audience, characteristics of a good security policy. Nature of security policies - data sensitivity policy, government agency IT security policy. Physical security - natural disaster, human vandals, interception of sensitive information.

References

1. C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 4th Edition, Pearson India, ISBN: 9788131727256.
2. Matt Bishop, *Computer Security: Art & Science*, 1st Edition, Pearson, ISBN: 0201440997.
3. William Stallings, *Cryptography and Network Security: Principles and Practice*, 6th Edition, Pearson India, ISBN: 9332518777.
4. Michael E. Whitman and Herbert J. Mattord, *Principles of Information Security*, 4th Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131516458.

CSS3E05d | Molecular Simulation and Modeling

Elective III: CSS3E05 Course Number: 3.5d Contact Hours per Week: 4

L	P	C
4	0	4

Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

- To understand application of simulation techniques to study molecular dynamics and derive properties.
- To learn and apply the statistical approaches and models for phylogenetic analysis and tree reconstruction.
- To understand the basis and nature of protein-protein interactions.
- To understand principles of docking simulations.

Course Outline

Unit I

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

Unit II

System modeling - concept, principles of mathematical modeling, static physical model, stochastic activities, continuous and discrete simulation. Discrete system simulation - probability concepts in simulation, random number generations and their testing, stochastic variable generation. Model execution - event driven versus time driven.

Unit III

Computational gene mapping - genetic mapping, gene expression, gene prediction methods, gene prediction tools, mutational analysis, introduction to restriction mapping and map assembly, mapping with restriction fragment fingerprints, Lander- Waterman statistics. Software Packages for Phylogenetic Analysis - PHYLogeny Inference Package (Phylip), Phylogenetic Analysis using Parsimony (PAUP) and Phylogenetic Analysis by Maximum Likelihood (PAML). Microarray technology - techniques for microarray data analysis - microarray databases. Scatter Plots,

Principal Component Analysis, Cluster Analysis, Applications of Microarray Technology.

Unit IV

Structural Modeling: Protein structure prediction - Prediction of protein secondary structure from the amino acid sequences. Prediction of three dimensional protein structure. Protein structure classification: Two major classification schemes - CATH and SCOP. Protein structure prediction: Steps involved in homology modeling. Protein- Protein Interactions: Prediction methods for Protein- Protein interactions. Protein- protein interaction Databases. Computer Assisted Drug Design (CADD): Protein based drug design cycle, drug discovery pipeline. Docking Simulations: Rigid docking and Flexible docking.

Unit V

Molecular Visualization: Visualization of protein structure, Methods of studying proteins, Proteomics databases, Protein family databases, PDB file format. Software tools for 3D molecular graphic visualization: Rasmol- basic operations and steps in Rasmol to visualize the molecule, advantages of Rasmol, advantages of Swiss- PdbViewer.

Text Books

1. Stephen Misener and Stephen A. Krawetz, *Bioinformatics: Methods and Protocols*, 1st Edition, Humana Press, ISBN: 1617371564.
2. Geoffrey Gordan, *System Simulation*, 2nd Edition, PHI, ISBN: 9788120301405.
3. Tamar Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide*, 2nd Edition, Springer, ISBN: 1461426502.
4. Narsingh Dev, System *Modelling with Digital Computer*, PHI, ISBN: 0138817898.
5. Andrew Leach, *Molecular Modelling: Principles and Applications*, Prentice Hall. 2nd Edition, ISBN: 81317286092001.
6. Prakash S Lohar, *Bioinformatics*, MJP publishers, Chennai, ISBN: 9788180940668.
7. H-D Holtje, *Molecular Modeling - Basic Principles and Applications*, 3rd Edition, Wiley-VCH, ISBN-13: 9783527315680.
8. Alan Hinchliffe, *Molecular Modelling for Beginners*, 2nd Edition, John Wiley and Sons Ltd, ISBN: 9780470513149.
9. N Cohen, *Guidebook on Molecular Modeling in Drug Design*, 1st Edition, ISBN :9780121782450
10. Masatoshi Nei and Sudhir Kumar, *Molecular Evolution and Phylogenetics*, Oxford University Press, ISBN: 0195135857.

References

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

CSS3E05e | Fundamentals of Big Data

Elective III: CSS3E05

L	P	C
4	0	4

Course Number: 3.5e
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

Introduction to Big Data – definition & importance of Big Data - four dimensions of Big Data - volume, velocity, variety, veracity – importance of big data – structured data, unstructured data - the role of a CMS in big data management - integrating data types into a big data environment - distributed computing and Big Data. Big Data stack – layer 0,1 and 2 – Big Data management – operational databases – relational databases – non relational databases – NoSQL - key-value pair databases
– document databases - columnar databases - graph databases - spatial databases.

Unit II

Big Data analysis - basic analytics - operationalized analytics - modifying business intelligence products to handle Big Data - Big Data analytics examples - Analytics solutions - text analytics - exploring unstructured data - understanding text analytics
- analysis and extraction techniques - the extracted information - text analytics tools for Big Data - custom applications for Big Data analysis – R Environment - Google Prediction API - Characteristics of a Big Data AnalysisFramework.

Unit III

NoSQL databases - types - Advantages over Relational Databases - MongoDB – introduction - MongoDB philosophy - the data model – designing the database – collections – documents - data types - the _id Field – indexes - viewing available databases and collections – opening a database - inserting data - querying for data – retrieving documents - aggregation commands - grouping results - conditional operators - specifying an array of matches – applying criteria for search - $slice - $size - $exists - $type - $elemMatch - $not (meta-operator) - update() - save() - $inc - $set - $unset - $push - $pushAll - $addToSet - removing elements from an array - atomic operations - modifying and returning a document atomically - renaming a collection - removing data - referencing a database - implementing index-related functions - min() and max().

Unit IV

Hadoop – history – components – HDFS - MapReduce Basics – origins of MapReduce - map function – reduce function – putting them together – Hadoop common components – application development in Hadoop – Pig and Pig Latin – Load – Transform – Dump and Store – Hive – Jaql – getting our data into Hadoop – basic copy data – Flume – Zookeeper – HBase – Oozie – Lucene – Avro.

Unit V

Understanding MapReduce - key/value pairs - the Hadoop Java API for MapReduce - the Mapper class - the Reducer class - the Driver class - writing simple MapReduce programs - Hadoop-provided mapper and reducer implementations - Hadoop-specific data types - the Writable and WritableComparable interfaces - wrapper classes - Input/output - InputFormat and RecordReader - OutputFormat and RecordWriter. Implementing WordCount using streaming - analyzing a large dataset - summarizing the UFO data - summarizing the shape data - a relational view on data with Hive - creating a table for the UFO data - inserting the UFO data - redefining the table with the correct column separator - creating a table from an existing file - SQL views.

References

1. Hurwitz, Alan Nugent, Fern Halper and Marcia Kaufman, *Big Data for Dummies*, ISBN: 9781118504222.
2. Eelco Plugge, Peter Membrey and Tim Hawkins, *The Definitive Guide to MongoDB: The NOSQL Database for Cloud and Desktop Computing*, 1st Edition, Apress, ISBN: 9781430230519.
3. Chris Elaton, Derk Deroos, Tom Deutsch, George Lapis and Pual Zikopoulos, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, 1st Edition, ISBN: B006UWBBO6.
4. Garry Turkington, *Hadoop Beginner's Guide*, Packt Publishing Ltd, ISBN: 1849517304.

CSS3E05f | Web Engineering

Elective III: CSS3E05 | **L** | **P** | **C**
Course Number: 3.5f | **4** | **0** | **4**
Contact Hours per Week: 4
Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure: None
Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

Modeling web applications – introduction – modeling specifics in WE – levels – aspects – phases of customizations – modeling requirements – hypertext modeling - hypertext structure modeling concepts – access modeling concepts. Web application design – web design from an evolutionary perspective – information design – software design – merging information design & software design – problems and restrictions in integrated web design – a proposed structural approach – presentation design – presentation of nodes and meshes – device independent development – approaches – interaction design – user interaction – user interface

organization – navigation design – deigning a link representation – designing link internals – navigation and orientation – structural dialog for complex activities – interplay with technology and architecture – functional design.

Unit V

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

References

1. Gerti Kappel, Birgit Proll, Siegried Reich and Werner Retschitzegger, *Web Engineering: The Discipline of Systematic Development of Web Applications*, John Wiley and Sons Ltd, ISBN: 9780470064894.
2. Roger S Pressman and David Lowe, *Web Engineering: A Practitioner's Approach*, 1st Edition, Tata Macgraw Hill Publications, ISBN: 9780073523293.
3. Leon Shklar and Rich Rosen, *Web Application Architecture: Principles, Protocols and Practices*, 2nd Edition, Wiley, ISBN: 047051860X.
4. Guy W Leeky-Thompson, *Just Enough Web Programming with XHTML, PHP, and MySQL*, 1st Edition, Cenagage Learning, ISBN: 159863481X.
5. Anders Moller and Michael Schwartzbach, *An Introduction to XML and Web Technologies*, 1st Edition, Pearson Education, New Delhi, 2009.
6. Chrits Bates, *Web Programming: Building Internet Applications*, 3rd Edition, Wiley India Edition, ISBN: 8126512903.

Semester IV

CSS4C01 – Principles of Software Engineering

Course Number: 4.2 **Contact Hours per Week:** 2 **Number of Credits:** 2

L	P	C
2	0	2

Number of Contact Hours: 30 Hrs
Prerequisite/Exposure: None **Course Evaluation:** 100% (Internal)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

Project story preparation – key deliverables – communicating with experts – forms of communication – presenting ideas – common problems faced by a research scholar – report writing.

References

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

CSS4C02 | Project Work

Course Number: 4.3 Contact Hours per Week: 8 Number of Credits: 8

L	P	C
		8

Number of Contact Hours: 60 Hrs.

Prerequisite/Exposure:

Course Evaluation: 25% (Internal) + 75% (External)

Objectives

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

Course Outline

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

Guide has to constantly monitor the works done by the student, imparting him/her the necessary inputs for the successful completion of the project work.

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

GUIDELINES FOR SUBMISSION OF REPORT

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

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

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

physical appearance of the dissertation should also conform to university standards.

The dissertation is to be prepared in tex format (either Latex or a suitable Windows tex variant). The format of the report is included in Appendix A.

Students are also encouraged to present their work in IT fest/conference/workshop/journal with the assistance and guidance of the supervisor. This should pave as a good start for the student in the art of publishing/presenting his/her work to the outside world. Due weightage is accommodated for publications out of the project work (Refer Section 7) in the final evaluation.

Semester IV | Elective IV CSS4E01 | List of Electives

CSS3E01a |Digital Image Processing

Elective IV: CSS4E01 Course Number: 4.1a Contact Hours per Week: 5 Number of Credits: 4

L	P	C
4	1	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None
Course Evaluation: 100% (Internal)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

Image enhancement - basic grey level transformation - histogram equalization –

image subtraction - image averaging - spatial filtering - smoothing, sharpening filters – Laplacian filters. Enhancement in the frequency domain – frequency domain filters - smoothing, sharpening filters - homomorphic filtering.

Unit IV

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

Unit V

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

References

1. Richard E Woods and Rafael C Gonzalez, *Digital Image Processing*, 3rd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131726959.
2. B. Chanda and D.D. Majumder, *Digital Image Processing and Analysis*, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120343255.
3. A.K. Jain, *Fundamentals of Digital Image Processing*, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120309294.
4. W.K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4th Edition, John Wiley, ISBN: 0471767778.
5. Milan Sonka, Vaclav Hlavac and Roger Boyle, *Image Processing Analysis and Machine Vision*, 3rd Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131518833.

CSS4E01b | Advanced Topics in Database Design

Elective IV: CSS4E01 Course Number: 4.1b Contact Hours per Week: 5 Number of Credits: 4

L	P	C
4	1	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None Course
Evaluation: 100% (Internal)

Objectives

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

Course Outline

Unit I

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

Unit II

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

Unit III

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

Unit IV

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

Unit V

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

References

1. Elmasri and Navathe, *Database Systems – Models, Languages, Design and Application Programming*, 6th Edition, Pearson India, ISBN: 8131792471.
2. Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, 3rd Edition, McGraw - Hill Education, ISBN: 9339213114.
3. Korth, Silberchatz and Sudarshan, *Database System Concepts*, 6th Edition, McGraw-Hill Education India Pvt. Ltd, ISBN: 9332901384.
4. Alexis Leon and Mathews Leon, *Database Management System*, 1st Edition, Vikas Publishers, ISBN: 8182092221.
5. Peter Rob and Coronel, *Database Systems, Design, Implementation and Management*, 5th Revised Edition, Course Technology, ISBN: 061906269X.
6. C J Date, *Introduction to Database Systems*, 8th Edition, Addison-Wesley, ISBN: 0321197844.

CSS4E01c | Software Development for Portable Devices

Elective IV: CSS4E01 Course Number: 4.1c Contact Hours per Week: 5 Number of Credits: 4

L	P	C
4	1	4

Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None Course
Evaluation: 100% (Internal)

Objectives

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

Course Outline

Unit I

Introduction to Mobile Web (HTML 5) - Semantic Elements – Structural Elements - Basic formatting tags - heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image. Different attributes like align, color, bgcolor, font face, border, size. Navigation Links using anchor tag - internal, external, mail and image links. Lists - ordered, unordered and definition, table tag, HTML5 form controls - form, input types – color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week, text, password, textarea, button, checkbox, radio button, select box, hidden controls, calendar, date, time, email, url, search. Datalist, keygen, output - Introduction to CSS3.

Unit II

jQuery – introduction - Adding jQuery to web pages – downloading – accessing from CDNs - jQuery syntax - jQuery selectors - event methods - ready(), click(), dblclick(), mouseenter(), mouseleave(), mousedown(), mouseup(), hover(), foucs(), blur() - effects – hide, show, fading, sliding, animation - callback functions – chaining - methods for changing and manipulating HTML elements and attributes - adding new elements/content - append(), prepend(), after(), before() – removing elements - remove(), empty() - manipulating CSS3 - dimensions of elements and browser window – traversing – ancestors, descendants, siblings.

Unit III

Introduction to Android and smart phones, Android architecture & virtual machine, mobile technology terminologies, setting up the environment, setting up emulators, Android fundamentals - activities and applications activity life cycles, activity stacks, activity states. Introduction to manifest, resources & R.java, assets, values – strings.xml - form widgets, views, layouts & drawable resources - XML layouts, linear layouts, relative layouts, table layouts, Android widgets, UI XML specifications events, bundles & intents - explicit intents implicit intents event broadcasting with intents event reception with broadcast receivers, adapters and data binding.

Unit IV

Files, content providers and databases - saving and loading files, SQLite databases - Android database design - exposing access to a data source through a content provider content provider registration native content providers, Android Debug Bridge (adb) tool, Linkify.

Unit V

Adapters and widgets, notifications, custom components threads running on UI thread, Worker thread handlers & runnable asyntask (in detail), playing audio and video, recording audio and video, using the camera to take and process pictures. Networking & location based services - live folders, using sdcards – reading and writing, XML parsing - JSON parsing - including external libraries in applications, Map-based activities, Maps via intent and Map activity GPS, location based services configuration, geocoding, accessing phone services (Call, SMS, MMS), network connectivity services, using Wifi & Bluetooth action bar tabs and custom views on action bars.

References

1. Terry Felke-Morris, Web *Development & Design Foundations with HTML5*, 7th Edition, Addison-Wesley, ISBN: 0133571785.
2. *Html 5 Black Book: Covers CSS3, Javascript, XML, XHTML, Ajax, PHP and Jquery*, Kogent Learning Solutions Inc, ISBN: 9350040956.
3. Kessler, *Programming HTML 5 Applications*, OReilly Media, ISBN: 9350235904.
4. Robin Nixon, *Html5 For Ios And Android: Beginner Guide*, 1st Edition, McGraw-Hill Education India Pvt .Ltd, ISBN: 101259003078.
5. Lauren Darcey and Shane Conder, *Android Wireless Application Development : Android Essentials (Volume 1)*, 3rd Edition, Pearson Education, ISBN: 9332518882.
6. Zigurd Mednieks, Rick Rogers, Lombardo John and Blake Meike, *Android Application Development*, 1st Edition, O'Reilly Meida,
7. Reto Meier, *Professional Android 2 Application Development*, 1st Edition, Wiley India Pvt Ltd, ISBN: 8126525894.

CSS4E01d | Storage Area Networks

Elective IV: CSS4E01 Course Number: 4.1d Contact Hours per Week: 5

L	P	C
4	1	4

Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None Course
Evaluation: 100% (Internal)

Objectives

- Understand Storage Area Networks (SAN) characteristics and components.
- Learn about the SAN architecture and management.
- Understand about designing and building SAN.

Course Outline

Unit I

Basic networking concepts and topologies - OSI reference model, common network devices, network topologies, MAC standards - need for storage networks – storage devices - techniques evolution - benefits of SANs - SAN components and building blocks - fibre channel basics - fibre channel topologies, fibre channel layers, classes of service SAN topologies.

Unit 2

SAN fundamentals - SAN operating systems software and hardware types of SAN technology - technology and configuration, high scalability and flexibility standards - storage management challenges - networked storage implementation challenges - storage subsystems for video services.

Unit III

Storage networking architecture storage in storage networking - challenges, cost and performance - Network in storage networking - fibre channel, emerging SAN interconnect technologies - basic software, advanced software, backup software implementation strategies.

Unit IV

Storage network management in-band management out-of-band management - SNMPHTTP - TELNET storage network management issues - storage resource

management - storage management, storage, systems and enterprise management integration.

Unit V

Designing and building a SAN - design considerations - business requirements - physical layout, placement, storage, pooling, data availability, connectivity, scalability, migration, manageability, fault tolerance and resilience - prevention of congestion – routability - backup and restoration - SAN security & iSCSI technology - basic security guidelines - implementing SAN security - backup and restoration in iSCSI technology - future of SANS.

References

1. Meeta Gupta, *Storage Area Network Fundamentals*, Cisco Press, ISBN: 158705065X.
2. John R. Vacca, *The Essential Guide to Storage Area Networks*, 1st Edition, Prentice Hall, ISBN: 0130935751.
3. Richard Barker and Paul Massiglia, *Storage Area Network Essentials: A Complete Guide to Understanding and Implementing SANs*, Wiley India Pvt Ltd, ISBN: 8126518588.
4. Tom Clark, *Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs*, 2nd Edition, Addison Wesley Professional, ISBN: 0321136500 .
5. Robert Spalding, *Storage Networks: The Complete Reference*, 1st Edition, Tata McGraw-Hill Education, ISBN: 0070532923.
6. Christopher Poelke and Alex Nikitin, *Storage Area Networks for Dummies*, 2nd Edition, ISBN: 9780470385135.
7. Ulf Troppens, Rainer Erkens, Wolfgang Mueller-Friedt, Rainer Wolafka and Nils Haustein, *Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE*, Wiley India Pvt Ltd, ISBN: 8126518324.

CSS4E01e | Semantic Web

Elective IV: CSS4E01

L	P	C
4	1	4

Course Number: 4.1e
Contact Hours per Week: 5
Number of Credits: 4
Number of Contact Hours: 60 Hrs
Prerequisite/Exposure: None Course
Evaluation: 100% (Internal)

Objectives

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

Course Outline

Unit I

Components – types – ontological commitments – ontological categories – philosophical background – knowledge representation ontologies – toplevel ontologies – linguistic ontologies – domain ontologies – semantic web – need – foundation – layers – architecture.

Unit II

Languages for semantic web and ontologies - web documents in XML – RDF - schema – web resource description using RDF - RDF properties – topic maps and RDF – overview – syntax structure – semantics – pragmatics - traditional ontology languages – LOOM - OKBC – OCML - Flogic Ontology Markup Languages – SHOE – OIL – AML – OIL – OWL.

Unit III

Ontology learning for semantic web - taxonomy for ontology learning – layered approach – phases of ontology learning – importing and processing ontologies and documents – ontology learning algorithms – evaluation.

Unit IV

Ontology management and tools - overview – need for management – development process – target ontology – ontology mapping – skills management system – ontological class – constraints – issues. Evolution – development of tools and tool suites – ontology merge tools – ontology based annotation tools.

Unit V

Applications - web services – semantic web services - security issues – current trends.

References

1. Asuncion Gomez-Perez, Oscar Corcho and Mariano Fernandez-Lopez, *Ontological Engineering: with examples from the areas of Knowledge Management, e- Commerce and the Semantic Web*, 1st Edition, Springer, ISBN: 1849968845.
2. Grigoris Antoniou and Frank van Harmelen, *A Semantic Web Primer*, The MIT Press, ISBN: 0262012103.
3. Liyand, *Introduction to the Semantic Web and Semantic Web Services*, Chapman, ISBN: 1584889330.
4. Alexander Maedche, *Ontology Learning for the Semantic Web*, Springer, 2002nd Edition, ISBN: 0792376560.
5. John Davies, Dieter Fensel and Frank Van Harmelen, *Towards the Semantic Web: Ontology – Driven Knowledge Management*, 1st Edition, Wiley, ISBN: 0470848677.
6. ieter Fensel, Wolfgang Wahlster, Henry Lieberman and James Hendler, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potenti al*, The MITPress, ISBN: 9780262562126.

CSS4E01f | Advanced Java Programming

Elective IV: CSS4E01

L	P	C
4	1	4

Course Number: 4.1f
Contact Hours per Week: 5
Number of Credits: 4
Number of Contact Hours: 60 Hrs.
Prerequisite/Exposure: None
Course Evaluation: 100% (Internal)

Objectives

- To learn the advanced features of Java programming language that equip the students to develop web based applications with RDBMS.

Course Outline

Unit I

RMI & Servlets - introduction, architecture, defining remote objects, creating stubs and skeletons, serializable classes, accessing remote objects, factory classes, dynamically loaded classes, RMI activation, registering remote objects.

Unit II

Servlets, generic servlet, servlets that access request headers, develop servlets that manipulate response headers, HTTP servlets, forms, HTTP protocols - configuring Tomcat Server, servlet context, servlet context listener, servelet chaining.

Unit III

JNDI & EJB - architecture, context initial context class, objects in a context, binding objects, accessing directory services, attributes and attribute interface modifying directory entities, creating directories entities. EJB roles, architecture, container, implementing a basic EJB object, implementing session beans, implementing entity bean, deploying an enterprise bean object.

Unit IV

Java Server Pages (JSP) - developing JSP pages, technology, syntax using scripting elements, syntax using the courier page directive, create and use JSP error pages, building reusable web presentation, components, JSP technology syntax using the include directive, JSP technology syntax using the jsp:include standard action, developing JSP Pages using custom tags, problem with JSP technology scriptlet code, given an existing custom tag library, develop a JSP page using the library, developing a simple custom tag, structure and execution of a custom tag in a JSP page, tag handler class for a simple empty custom tag, custom tag that includes its body in the contour of the HTTP response, tag library description for a simple, empty custom tag.

Unit V

Hybernate - ORM overview - Hibernate overview, environment, configuration, sessions, persistent class - mapping files - mapping types - examples - O/R mappings - annotations - Hibernate Query Language - Hibernate criteria - queries - Hibernate Native SQL, caching, batch processing, interceptors.

References

1. Jason Hunter and William Crawford, *Java Servlet Programming*, 2nd Edition, O'Reilly Media, ISBN: 0596000405.
2. Karl Moss, *Java Servlets*, McGraw-Hill, ISBN: 0074637398.
3. Barry Burd, *JSP: JavaServer Pages*, IDG Books, ISBN: 0764535358.
4. Prashant Sridharan, *Javabeans Developer's Resource*, ISBN: 0138873089.
5. Chuck Cavaness, *Programming Jakarta Struts*, 2nd Edition, O'Reilly Media, ISBN: 0596006519.
6. Madhusudhan Konda, *Just Hibernate: A Lightweight Introduction to the Hibernate Framework*, Oreilly Meida, ISBN: 9781449334376.

APPENDIX A

CSS4C02 | PROJECT WORK | GUIDELINES FOR PROJECT REPORT

PROJECT REPORT LAYOUT

COVER PAGE & FIRST PAGE

TITLE HERE

A PROJECT REPORT

<<College/Dept Emblem>>

Name of the Department
Name of the Institution
(Affiliated to the CHRIST COLLEGE(AUTONOMOUS), IRINJALAKUDA) Address

MONTH YEAR

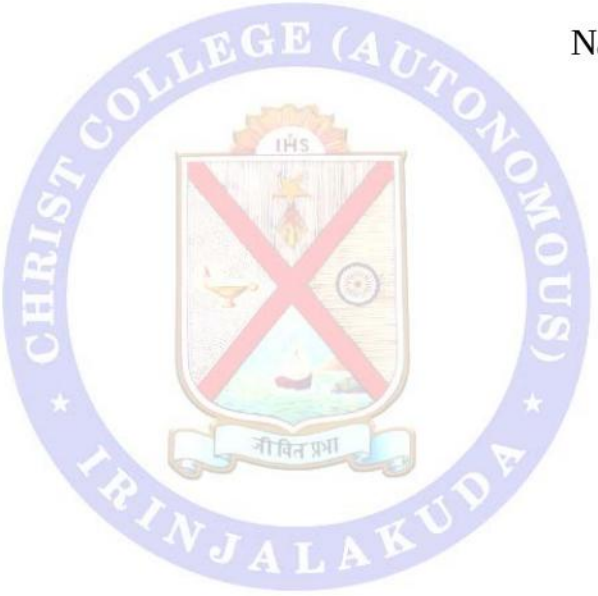
ACKNOWLEDGEMENT

Acknowledgments

I would like to thank THE ALMIGHTY's mercy towards me over the years……..

....

Date: Name of the Student



DECLARATION BY THE STUDENT

DECLARATION

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

Signature :

Place : Name:

Date : Reg. No.:

DECLARATION BY THE STUDENT

DECLARATION

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

Signature :
Place : Name:
Date : Reg. No.:

CERTIFICATE FROM GUIDE & HOD

CERTIFICATE

This is to certify that the project report entitled TITLE HERE!!! submitted by <<Name of the Student>> (Register Number:) to CHRIST COLLEGE(AUTONOMOUS), IRINJALAKUDAfor the award of the degree of Master of Science (M.Sc.) in Computer Science is a bonafide record of the project work carried out by him/her under my supervision and guidance. The content of the report, in full or parts have not been submitted to any other Institute or University for the award of any other degree or diploma.

Signature

Name Project Guide
Designation

Signature

Name of the HOD
Designation

Place :
Date :

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

Examiners:

1.

2.

iv

LIST OF CONTENTS

Contents

v

ABSTRACT

Abstract

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

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

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

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

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

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

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

List of Figures

List of Tables

Chapter 1

Introduction

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

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

Chapter 2

Problem Definition and Methodology

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

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

Chapter 3

Requirement Analysis and Specification

This chapter includes the following subsections.

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

Chapter 4

System Design

This chapter includes the following subsections.

1. Users of the System
2. Modularity Criteria
3. Architecture Diagrams (whichever of the following if applicable)
    a. DFD
    b. UML Diagrams
    c. Flowchart
4. User Interface Layout
5. Structure of Reports Being Created
6. Database Design
    a. List of Entities and Attributes
    b. E R Diagram
    c. Structure of Tables

Chapter 5

Implementation

This chapter is about the realisation of the concepts and ideas developed earlier. It can also describe any problems that may have arisen during implementation and how you dealt with them.

Do not attempt to describe all the code in the system, and do not include large pieces of code in this section. Instead pick out and describe just the pieces of code which, for example:

- are especially critical to the operation of the system;
- you feel might be of particular interest to the reader for some reason;
- illustrate a non-standard or innovative way of implementing an algorithm, data structure, etc.

You should also mention any unforeseen problems you encountered when implementing the system and how and to what extent you overcame them. Common problems are:

- difficulties involving existing software, because of, e.g.,
    - its complexity,
    - lack of documentation;
    - lack of suitable supporting software;
    - over-ambitious project aims.

A seemingly disproportionate amount of project time can be taken up in dealing with such problems. The Implementation section gives you the opportunity to show where that time has gone.

Complete source code should be provided separately as an appendix. This

chapter includes the following subsections.

1. Brief description about the Tools/Scripts for Implementation
2. Module Hierarchy

3. Coding
4. Problems Encountered

Chapter 6

Testing & Implementation

This chapter includes the following subsections.

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

Chapter 7

Conclusion

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

There are a couple of rules for this chapter:

- All material presented in this chapter must have appeared already in the report; no new material can be introduced in this chapter (rigid rule of technical writing).
- Usually, you would not present any figures or tables in this chapter (rule of thumb).

Conclusions section can have the following (typical) content. These contents must **not** be given in bulleted format.

- Re-introduce the project and the need for the work though more briefly than in the introduction.
- Reiterate the purpose and specific objectives of your project.
- Recap the approach taken similar to the road map in the introduction. However, in this case, you are re-capping the data, methodology and results as you go.
- Summarize the major findings and recommendations of your work.

Future Enhancements

Identify further works that can be added to make your system to meet the challenges of tomorrow. You can also include whatever requirements you could not fully due to the scarcity of time/resources.

Bibliography

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

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

Publications out of The Project Work

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

GENERAL INSTRUCTIONS

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

.

APPENDIX B

CSS4C02 | PROJECT WORK | A SAMPLE EVALUATION

Scheme for internal evaluation is as follows.

Components	Weightage
Monthly progress	4
Regularity	1
Total	**5**

Suppose that we plan 6 phases for monthly review. Evaluation criteria for each phases is as shown below. Recall that this can be set conveniently by the EC.

Phase	Component	Grade Points
Phase I	Relevance of the Topic, Statement of Objectives, Methodology	20
	Quality of Presentation	2
	Quality of Interim Report	**5**
	Total	**27**
Phase II	Quality of Literature Survey/Product Review	20
	Quality of Presentation	2
	Quality of Interim Report	5
	Total	**27**
Phase III	Quality of Analysis Phase	20
	Quality of Presentation	2
	Quality of Interim Report	5
	Total	**27**
Phase IV	Quality of Design Phase	20
	Quality of Interim Report	5
	Quality of Presentation	2
Phase V	Quality of Implementation/Simulation	50
	Quality of Interim Report	5
	Quality of Presentation	5
	Total	87
Phase VI	Quality of Testing/Result Analysis	20

	Quality of Contributions	20
	Identification of Future Work	8
	Quality of Project Report	25
	Publications/Presentations out of the Project Work	10
	Demonstration of the Project Work	10
	Total	93
Total		**288**

Each of the phases such as Analysis Phase can be conveniently subdivided into subsections such as DFD Level 0, DFD Level 1, DFD Level 2, ER Diagram/Class Diagram. Other components in each phase can also be subdivided appropriately, if required.

Assume that a student got 250 points after 6 phases of evaluation. Also assume that his/her regularity performance worth 4 points. Now the final grade for the student is calculated as ((250/72) X 4 + 4 X 1)/5 = (3.47 X 4 + 4)/5 = 3.57 (A grade).

APPENDIX C
MODEL QUESTION PAPERS

Model question papers for few papers are included as a guideline for the faculty/subject experts for question papers/students.

No	Course Code	Course Name
1	CSS2C01	Design and Analysis of Algorithms
2	CSS2C02	Operating System Concepts
3	CSS2E05a	Computer Graphics
4	CSS3C01	Advanced Database Management System
5	CSS3C02	Principles of Compilers
6	CSS2C03	Computer Networks
7	CSS3C03	Object Oriented Programming Concepts
8	CSS3E04c	Cryptography and Network Security
9	CSS3E05c	System Security
10	CSS3E05d	Molecular Simulation and Modeling
11	CSS4E01c	Software Development for Portable Devices

CSS2C01 | Design and Analysis of Algorithms

Time: Three Hours Maximum: 36 Weightage

Part A

Answer all questions

Each question carries weightage of 1

1. Write a note on analyzing complexity of an algorithm.
2. Write a note on Brute Force method.
3. Write a note on PRAM Model.

   Big Theta Ratio Theorem.
4. What is meant by Greedy approach?
5. Write a note on Euler Tour Technique.
6. What do mean by Parallel Prefix Computation?
7. What is the use of Iteration method?
8. Write a note on Nondeterministic Polynomial time algorithm.
9. Write a note on Brent's theorem.
10. What do you mean by ERCW?
11. Define Amdahl's Law.

(12X1=12 weightage)

Part B

Answer any 6 questions, Each question carries 2 weightage

12. Solve the recurrence $\mathbf{T(n) = 3T(n/2) + n^2}$
13. How can we solve Sum of subsets problem using Backtracking?
14. What do mean by Recursion Tree Method?
15. Let $\mathbf{f(n) = 4n^3 + 4n^2 + 3n + 2}$, Show that $\mathbf{f(n) = \Omega(n^3)}$
16. Distinguish between Kruskal's algorithm and Prim's algorithm.
17. Solve the recurrence $\mathbf{T(n) = 2T(\sqrt{n}) + \log n}$
18. Let $\mathbf{f(n) = 2 * 3^n + 5n^2 + 3n}$, Show that $\mathbf{f(n) = \Theta(3^n)}$
19. How can we solve Knapsack problem using Branch-and-Bound technique?
20. Let $\mathbf{T(n) = T(n-1) + n^4}$, Show that $\mathbf{T(n) = O(n^5)}$.

(6X2=12 weightage)

Part C

Answer any 3 questions, Each question carries 4 weightage

21. Show that Strassen's matrix multiplication algorithm is faster than standard matrix multiplication algorithm.
22. Explain Dynamic Programming with an example.
23. The recurrence $\mathbf{T(n) = 7T(n/2) + n^2}$ describes the running time of an algorithm **A**. A competing algorithm $\mathbf{A^I}$ has a running time of $\mathbf{T^I(n)= kT^I(n/4)+n^2}$. What is the largest integer value for **k** such that $\mathbf{A^I}$ is asymptotically faster than **A**?
24. Prove that Hamiltonian Cycle is NP Complete.
25. Prove that the runtime complexity of merge sort in best case is $\mathbf{\theta(n \log n).}$
26. Can the master method be applied to the recurrence $\mathbf{T(n) = 2T(n/2) + n \log n}$? Why or Why not?

(3X4=12 weightage)

CSS2C02 | Operating System Concepts

Time: Three Hours Maximum: 36 Weightage

Part A

Answer all questions

Each question carries weightage of 1

1. Give any two objectives of Operating Systems.
2. Define multi threading.
3. List the requirements formutual exclusion.
4. Give the deadlock recovery strategy.
5. Differentiate between reusableresource and consumable resource.
6. What is the difference between internal and external fragmentation?
7. What is mean by demand paging?
8. Differentiate between internal and external fragmentation.
9. What is mean by priority inversion?
10. Explain best fit and first fit algorithms formemory allocation.
11. Explain the concept of semaphore.
12. Give the advantages of Remote Procedure Call.

Part B

Answer any six questions

Each question carries a weightage of two

13. What is the difference between mode switch and process switch?
14. Explain two categories of thread implementation.
15. Explain monitors.
16. Explain the conditions for Deadlock. How deadlock can be described in terms of resource allocation graph.
17. Explain thread scheduling.
18. What are the characteristics of Real Time Operating Systems?
19. Explain about any two concurrency mechanisms in Unix.
20. Explain trashing.
21. How does client/serverdifferfrom otherdistributed processing solution.

Part C

Answer any three questions

Each question carries a weightage of four

22.

a. Explain reasons forprocess termination

b. Draw the UNIX process state transition diagram and explain.

23. Consider the following snapshot forasystem

Process	Allocation				Maximum				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	2	3	5	6				
P4	0	0	1	4	0	6	5	6				

Answerthe following questions using the banker's algorithm

What is the content of the matrix Need.
Is the system is a safe state
If a request from P1 arrives for(0, 4, 2, 0) can the request be granted immediately?

24. Explain page replacement algorithms.
25. Explain real time scheduling algorithms.
26. Explain (a) readers writers problem (b) Dining Philosopher's problem.
27. Explain distributed message passing in asingle system.

CSS2E05a |Computer Graphics

Time: Three Hours Maximum: 36 Weightage

Part A

Answer **all** *questions.*

Each question carries 1 *weightage.*

1. Define aspect ratio?
2. Define window and viewport?
3. What is clipping?
4. Give matrix representation for 3D reflection.
5. What do you mean by homogeneous co-ordinates? How these co-ordinates are useful in transformations?
6. What do you mean by texture mapping?
7. What do you mean by rendering?
8. Explain caveliar and cabinet projections?
9. Explain interpolation and approximation splines.
10. What are the disadvantages of DVST?
11. What do you mean by scan conversion?
12. What are the properties of a Bezier curve?

(12 x 1 = 12 weightage)

Part B

Answer any six questions.

Each question carries 2 weightage.

13. Write a simple OpenGL program and explain its working.
14. Explain about monochrome video displays.
15. How is computer graphics useful in education and training? Explain.
16. Explain geometric continuities and parametric continuities. What is the condition for smoothly joining curve segments?
17. Describe the shadow mask technique. What are its limitations?
18. Describe classification of visible surface detection algorithm.
19. What do you understand by the term vanishing points? Which type of projection is associated with vanishing points?
20. What are plasma panels and how do they work?
21. Describe scan line seed fill algorithm for polygon filling.

(6 x 2 = 12 weightage)

Part C

Answer any **three** *questions*

Each question carries 4 *weightage*

22. Explain briefly Cohen-Sutherland line clipping algorithm.
23. Explain the working of CRT with the help of a neat diagram.
24. Find the transformation matrix that transforms the given square ABCD to half its size with centre still remaining at the same position. The co-ordinates of the square are A(1,1), B(3,1), C(3,3), D(1,3) and centre at (2,2). Also find the resultant co-ordinates of square.
25. Discuss sampling and aliasing techniques.
26. Describe the Bresenham's algorithm to draw a circle whose centre is origin and radius is 7.
27. Give a detailed account of orthogonal projection.

(3 x 4 = 12 weightage)

CSS3C01|Advanced Database Management System

Time: Three Hours Maximum: 36 Weightage

Part A

Answer **all** *questions.*

Each question carries 1 *weightage.*

1.What does logical Data Independence in the context of DBMS.

2.What is meant by data abstraction.

3.Explain derived attribute with asimple example.

4.What are the objectives of Database Normalisation.

5.How do you represent multivalued attributes in ERdiagram.

6.What are the different types of anomalies that may occur while designing adatabase.

7.List the DDL commands in SQL

8.What are the aggregate functions in SQL.

9.List the basicproperties of atransaction

10..What do you mean by serializability in transaction management.

11.Discuss the concept of encapsulation and how it is used to create abstract data types,

12.Differentiatedatabase schemaand state.

Part B

Answer any ***six*** *questions*

Each question carries a weightage of 2

13. Explain the three schemaarchitecture in Database Management System.

14.Consider the relation R(A,B,C,D,E) with the following dependencies.

{AB->C,CD->E,DE->B}.Find the candidate key forthis relation.

15.Discuss how the entity integrity and referential integrity are implemented using SQL.

16.Discuss with example why concurrency control is needed in database transaction.

17.What are the main differences between designing arelationaldatabase and an object database.

18.Explain fragmentation and replication in

DDBMS. 19.Explain recursive relation with suitable

example.

20.Between the properties dependency preservation and losslessness, in database design,which one must be definitely satisfied.Why.

21.Define the term deadlock and starvation with suitableexample.

Part C

Answer any ***three****questions*

Each question carries a weightage of 4

22.Discuss the various types of join operations in relationalalgebra. How are the outer join operations different from join operations.

23.Define Boyce-Codd normal form.How is it different from third normal form.Why is it considered a strongerform of 3NF.

24.What is aview in SQL.Discuss the problems related to view update and how views are implemented.

CSS3C02 |Principles of Compilers

Time: Three Hours Maximum: 36 Weightage

Part A
Answer all questions
Each question carries 1 weightage

1. What is a Compiler?
2. What are the three phases of analysis of a source program?
3. Define lexical analysis.
4. Define symbol table.
5. What is meant by backpatching?
6. Define input buffering.
7. Define regular expression.
8. What is meant by a parser?
9. Define quadruple.
10. Write a note on peephole optimization.
11. What do you mean by Lexeme?
12. Define ambiguity.

(12X1=12 weightage)

Part B
Answer any 6 questions
Each question carries 2 weightage

13. What is meant by error recovery strategies?
14. Write a note on parse tree and derivation.
15. Define LL(1) Grammar.
16. Explain briefly about top down parsing.
17. What are the two standard allocations in run- time storage management?
18. What is algebraic simplification?
19. What is meant by grouping of phases?
20. Define semantic analysis.
21. What are the different types of storage allocation strategies?

(6X2=12 weightage)

Part C
Answer any 3 questions.
Each question carries 4 weightage

22. What are the phases of a Compiler? Explain in detail with the support of a diagram.
23. Find, if the string aaa bbb ccc can be derived from the productions

    S □ ABSc, S □ Abc, BA □ AB, Bb □ bb, Ab □ ab and Aa□aa.
24. How can we convert an NFA to a DFA? Explain with an example.
25. What do you mean by Syntax analysis? Explain in detail.
26. Describe the process of creating symbol table entries.
27. Draw and explain the translation of the statement **k := i + j * 1024**

CSS2C03 |Computer Networks

Time: Three Hours Maximum: 36 Weightage

Part A
Answer all questions
Each question carries 1 weightage

1. Define Computer networks. Explain different types of networks.
2. Explain packet switching.
3. Differentiate between Frequency Division Multiplexing and Time Division Multiplexing.
4. Write a short note on HTTP.
5. Explain DNS.
6. What is mean by IP addressing?
7. Explain parity check for error detection.
8. Write a short note on internet.
9. Compare switch and a router.
10. Explain PPP.
11. What is a firewall?
12. Write a short note on authentication.

(12X1=12 weightage)

Part B
Answer any 6 questions
Each question carries 2 weightage

13. Explain different LAN topologies with the help of examples and diagrams.
14. How mail is delivered between sender and receiver using SMTP?
15. Explain routing. Differentiate between unicast and multicast routing.
16. Differentiate between pure Aloha and slotted Aloha.
17. Explain address resolution in ARP.
18. Compare IPv4 and IPv6.
19. Explain the services provided by transport layer.
20. Explain two different transmission mediums for communication.
21. Explain how FTP works.

(6X2=12 weightage)

Part C

Answer any 3 questions.

Each questions carries 4 weightage

22. Explain TCP/IP reference model. Also explain significance of each layer with the help of diagram.
23. Compare TCP and UDP with adequate diagrams.
24. Explain Cyclic Redundancy Check with example.
25. Explain CSMA and CSMA/CD.
26. Explain ATM in detail.
27. Explain data encryption and decryption in detail.

(3X4=12 weightage)

CSS3C03 | Object Oriented Programming Concepts

Time: Three Hours Maximum: 36 Weightage

Part A

Answer all questions

Each question carries 1 weightage

1. Define Encapsulation.
2. What is dynamic initialization?
3. What are inner classes?
4. What are the uses of *this* keyword?
5. List the restrictions of static methods.
6. What is synchronization?
7. What is the use of *finally* block?
8. List the advantages of java swing over AWT.
9. What are components and container?
10. What is a layout manager?
11. What is a socket?
12. What is a resultset? (12 X 1 = 12 weightage)

Part B

Answer any 6 questions

Each question carries 2 weightage

13. List the features of java.
14. Explain with an example, how run time polymorphism is implemented in java.
15. Differentiate method overloading and method overriding.
16. What is a Package? Explain how packages control the visibility of variables.
17. What are streams and stream classes?
18. What is an Exception? What is the difference between *throw* and *throws* statements.
19. Explain Delegation Event Model for event handling in java.
20. What are adapter classes? Explain how it helps in event handling.
21. Explain the architecture of JDBC. (6 X 2 weightage)

Part C

Answer any 3 questions.

Each question carries 4 weightage

22. What are the different control statements available in Java?
23. a) What is an interface? Explain how it differs from an abstract class.
    b) Explain how interfaces can be extended.
24. a)What is multithreading?
    b) Write a program to create two threads - even and odd that display even and odd numbers within a range.
25. a) What is an Applet? List the applet life cycle methods.
    b) Write an applet to accept two integers as parameter passed from the html file and display their sum, difference and product on the applet window.
26. Explain the different Statement interfaces available in JDBC.
27. Write a note on various UML diagrams.

(3X4=12 weightage)

CSS3E04c| Cryptographyand Network Security

Time: Three Hours Maximum: 36 Weightage

Part A
Answer all questions
Each question carries 1 weightage

1. Define Computer security.
2. What is masquerade?
3. Name the five ingredients of symmetric encryption.
4. Elaborate MAC.
5. What are the six ingredients of public-key encryption scheme?
6. Give the authentication service exchange sequence to obtain ticket granting ticket of Kerberos 4.
7. Define ticket lifetime.
8. What are the elements of communication that are encrypted while using HTTPS.
9. What is random padding?
10. Give an example of criminal enterprise.
11. What is rule based anomaly detection?
12. What is a honeypot?

(12X1=12 weightage)

Part B
Answer any 6 questions
Each question carries 2 weightage

13. Describe the three objectives of computer security.
14. Briefly describe stream cipher structure.
15. Explain RSA public key encryption algorithm .
16. What properties must a hash function have to be useful for message authentication?
17. How would you describe key distribution using asymmetric encryption?
18. Consider a one-way authentication technique based on asymmetric encryption:

    A—->B: ID_A

    B——>A:R_1

    A —-> B: E(PR_a, R_1)

    a. Explain the protocol.

    b. What type of attack is this protocol susceptible to?

19. Describe SSL Record protocol.
20. List and briefly define the parameters that define an SSL session state.
21. List out intrusion techniques.

(6X2=12 weightage)

Part C

Answer any 3 questions.

Each questions carries 4 weightage

22. Explain in detail Feistel Cipher structure
.
23. Describe secure hash functions.
24. Write notes on X 509 certificates.
25. Explain DDoS.
26. Describe types of malicious softwares.
27. List and describe the SSH protocols.

(3X4=12 weightage)

CSS3E05c | System Security

Time: Three Hours Maximum: 36 Weightage

Part A
Answer all questions
Each question carries 1 weightage

1. What is vulnerability?
2. Differentiate between modification and fabrication.
3. What is "penetrate and patch"?
4. Differentiate between transient and resident viruses.
5. Define relocation.
6. Give a short description of single sign-on.
7. List the qualities of security and trustiness.
8. What are the security requirements of a database?
9. Give the essential features of a sensitivity label of integrity lock.
10. List all the groups of personnels who should have representatives in a security planning team.
11. What is an incident response plan?
12. Define risk impact.

(12X1=12 weightage)

Part B
Answer any 6 questions
Each question carries 2 weightage

13. Explain vulnerability, threats, attacks and controls.
14. Briefly describe different classes of computer criminals.
15. Explain incomplete mediation and it's security implications.
16. What is paging? How can paging be used in combination with segmentation?
17. What are the various types of intermediate code representations?
18. Explain file protection mechanisms.
19. Describe hazard analysis techniques.
20. How would you explain contingency planning?
21. Give a brief note on assurance management in trusted operating systems.

(6X2=12 weightage)

Part C
Answer any 3 questions.
Each questions carries 4 weightage

22. Explain Bell La Padula model of security .
23. What are the controls against program threats? 24.Explain the design of a trusted operating system.
25. Explain inference problem in database security.
26. Describe sensitive data is managed in a secure database.
27. Explain the steps of risk analysis.

(3X4=12 weightage)

CSS3E05d | Molecular Simulation and Modeling

Time: Three Hours Maximum: 36 Weightage

Part A
Answer all questions
Each question carries 1 weightage

1. What is molecular modeling?
2. Distinguish physical and statistical model?
3. What is a static variable?
4. What is a PDB file?
5. Write any two models available in molecular visualization.
6. What is central dogma of molecular genetics?
7. What is a coding region?
8. What is Open Reading Frames?
9. Write any to software gene prediction tools?
10. What is Molecular phylogenetic tree?
11. What is a Dendrogram?
12. What is the use of scatter plot in Microarray?

(12X1=12 weightage)

Part B
Answer any 6 questions
Each question carries 2 weightage

13. Distinguish between event driven versus time driven model execution
14. Write about discrete and continues simulation models.
15. What is genetic mapping?
16. What is gene prediction? Write any two methods.
17. Explain maximum likelihood approach in phylogenetic analysis
18. Briefly write the basic concept of microarray technology and its use.
19. Write short note on Rasmol and Swiss PDB viewer.
20. Distinguish between rigid and flexible docking.

(6X2=12 weightage)

Part C
Answer any 3 questions.
Each question carries 4 weightage

21. Briefly explain different models types in molecular modeling.
22. Explain gene expression in terms of protein synthesis.
23. Explain different method phylogenict analysis.
24. Different steps in Homology modeling.
25. Briefly write about drug discovery pipeline.

(3X4=12 weightage)

CSS4E01c | Software Development for Portable Devices

Time: Three Hours Maximum: 36 Weightage

Part A
Answer all questions
Each question carries 1 weightage

1. What are hidden controls?
2. What is datalist in HTML 5?
3. What is output element in HTML 5?
4. What is jQuery?
5. What is a selector element?
6. What is meant by Intents?
7. What is activity stack?
8. What is Linkify?
9. List any two types of native adapters?
10. What is android manifest?
11. Define AsyncTask?
12. What is geocoding?

(12X1=12 weightage)

Part B
Answer any 6 questions
Each question carries 2 weightage

13. List the formatting tags available in HTML 5?
14. What are the different modules in CSS3?
15. Write a short note on jQuery events.
16. Explain jQuery traversing with an example.
17. List the uses of Content providers?
18. Discuss the android activity life cycle.
19. What is android debug bridge?
20. Discuss the features of any android database.
21. Explain location based services and its components in the context of android application?

Part C
Answer any 3 questions.
Each question carries 4
weightage(6X2=12 weightage)

22. List and explain the different new form element types in HTML 5?
23. Write a short note on jQuery effects.
24. Discuss how phone services are accessed in android applications.
25. Discuss the components of an Android application.
26. What is a Layout? Enumerate various layouts available for android development.
27. Explain the XML and JSON parsing in Android.

(3X4=12 weightage)