KANNUR UNIVERSITY

(Abstract)

New Generation Course in M.Sc. Computer Science with Specialization in Artificial Intelligence Programme under Credit Based Semester System, at Govt. College, Thalassery, Chokli - Scheme, Syllabus and Pattern of Question Papers of Core and Generic Elective Courses - with effect from 2020 admission -implemented orders issued.

ACADEMIC C SECTION

Acad/C2/16584/NGC/2021 Dated: 22.02.2021

Read:-1. G.O(Ms) No.389/2020/HEDN dated 05.11.2020

2. Minutes of the meeting of Syndicate held on 17.11.2020 vide item No.2020.550

3. U.O No. Acad.A3/389/NEW COURSES/ 2020-21 dated 23.12.2020

4. U.O No Acad.C1/11460/2013 dated 12.03.2014

5. Minutes of the meeting of Curriculum Syllabus Monitoring Committee held on 20.11.2020

6. U.O No. Acad/C2/2408/2020 dated 27.11.2020

7. Syllabus submitted by the convenor Expert Committee dated 12.02.2021

ORDER

1. As per read (1) above, sanction was accorded by the Government to start New Generation UG/PG Courses in 15 Govt and Aided Colleges under Kannur University, during the academic year 2020-21.
2. Subsequently, the meeting of the Syndicate as per paper read (2) above resolved to start the newly Sanctioned UG & PG Programmes in Govt/Aided Colleges/University Departments from the academic year 2020-21.
3. Accordingly, provisional affiliation was granted for conducting the New Generation Course M.Sc. Computer Science with Specialization in Artificial Intelligence programme at Govt. College, Thalassery, Chokli, Kannur in the academic year 2020-21, as per paper read (3).
4. Further, the Curriculum Syllabus Monitoring Committee, as per paper read (5) above, resolved to follow the the existing Regulation for PG programmes in affiliated colleges under Credit Based Semester System implemented w.e.f 2014 admission as per read (4), for the New Generation Courses also. An Expert Committee was constituted for preparing the draft curriculum, Syllabus of New Generation Courses, by conducting two days' workshop as per paper read (6).
5. Accordingly, the scheme, Syllabus & Pattern of Question papers of M.Sc. Computer Science with Specialization in Artificial Intelligence programme CBSS prepared by the Expert Committee was submitted by the Convenor of Curriculum Syllabus Monitoring Committee as per paper read (7), for implementation w.e.f 2020 admission at Govt. College, Thalassery, Chokli, Kannur.
6. The Vice Chancellor after considering the matter in detail and in exercise of the power of the Academic Council conferred under Section 11(1) Chapter III of Kannur University Act 1996 accorded sanction to implement the Scheme, Syllabus and Pattern Question Papers of Core, and Generic Elective Courses of M.Sc Computer Science with Specialization in Artificial Intelligence programme(CBSS), at Govt. College, Thalassery, Chokli, Kannur, with effect from 2020 admission, subject to reporting to the Academic Council.
7. The Scheme, Syllabus & Pattern of Question Papers of M.Sc. Computer Science with

Specialization in Artificial Intelligence programme CBSS are uploaded in the University website. (www.kannuruniversity.ac.in).

8. Orders are issued accordingly.

Sd/-

BALACHANDRAN V K

DEPUTY REGISTRAR (ACAD)

For REGISTRAR

To: The Principal, Govt. College,Thalassery
Chokli,Kannur

Copy To: 1.The Examination Branch (through PA to CE)
2. PS to VC/ PA to PVC/ PA to Registrar
3. DR/AR I Academic/Academic A Section
4. The Computer Programmer (for uploading website)
5. SF/DF/FC

Forwarded / By Order

SECTION OFFICER

KANNUR UNIVERSITY
CIVIL STATION P.O.
KANNUR
Pin-670 002
THAVAKKARA

KANNUR UNIVERSITY

M.SC. COMPUTER SCIENCE
with Specialization in Artificial Intelligence

under

KUCBSS

Scheme and Syllabus

2020-21

Programme Specific Outcomes (PSO):

After the successful completion of two year MSc Computer Science program in Artificial Intelligence, a student should be capable of:

- Communicate computer science concepts, designs, and solutions effectively and professionally.
- Apply knowledge of computing to produce effective designs and solutions for specific problems.
- Use software development tools, software systems, and modern computing platforms
- An understanding of the fundamental design mechanisms, algorithms, and state-of-the-art system architectures in AI, along with knowledge of the core areas of AI, including machine learning, natural language processing, and computer vision.
- The ability to apply creative thinking, algorithmic design skills, and coding skills to build modern AI systems.
- The ability to convert descriptions of abstract AI challenges into descriptions of specific AI project requirements.

Expert Committee

1. Dr. Thomas Monoth
   (Convener)
   Associate .Professor,
   Department of Computer Science
   Mary Matha College
   Mananthavady

2. Dr. Sabu M K,
   Professor,
   Department of Computer Applications,
   CUSAT, Cochin.

3. Dr.Rajesh R,
   Associate .Professor,
   Department of Computer Science,
   Central University of Kerala.

4. Dr. Sanil Sankar
   Assistant .Professor
   Department of IT
   Kannur University

KANNUR UNIVERSITY

M Sc COMPUTER SCIENCE

(Credit Based Semester System)

Regulations, Curricula, Syllabus and Scheme of Evaluation

(With Effect from 2014 admission)

REGULATIONS

1. **Duration** of the M. Sc. (Computer Science) programme shall be 2 years, divided into 4 semesters. Each semester shall have 90 working days.The maximum period of completion is eight semesters (4 years).

2.Eligibilityfor admission: As announced by the University from time to time.

3. Programme Structure

3.1 **Attendance*:*** The minimum attendance required for each course shall be 75% of the total number of classes conducted for that semester. Those who secure the minimum attendance in a semester alone will be allowed to register for the End Semester Examination. Condonation of shortage of attendance may be granted as per Kannur University PG regulation.

3.2 ***Credits:*** The total minimum credits, required to complete M. Sc. Computer Science programme is 80 in which minimum credits required for core (including practical and project) courses is 60 and for Elective courses is 12.

3.3 **Theory and Practical courses**

The evaluation scheme for each Theory and Practical courses except MCS3C16 Research Methodology shall contain two parts; (a) Continuous Assessment (CA) and (b) End Semester Evaluation (ESE). 20% marks shall be given to CA and the remaining 80 % to ESE. For MCS3C16 Research methodology the evaluation is 100% internal and shall follow the distribution applicable to theory CA.

CONTINUOUS ASSESSMENT (CA)

Theory : The components oftheory evaluation are as follows:

	COMPONENTS	% OF MARKS
i	**Test papers**	**40%**
ii	**Assignment**	**20%**
iii	**Case Study / Seminar / Viva**	**20%**
iv	**Attendance**	**20%**

i. *Test Papers*: There shall be a minimum of two test papers to be conducted for each course. If more than two test papers are conducted, then two best scores shall be taken for the award of IA marks. The dates of test papers shall be announced well in advance and the marks should be displayed in the notice board.

ii. *Assignments:* One or more assignments (including practical assignments) shall be given for each course. The mode of assessment of the assignments shall be decided by the faculty concerned with due approval from the department council and shall be declared at the beginning of the semester. (It is suggested that to the extent possible, give individual assignments and also conduct short viva based on the assignment submitted).

iii. *Case study / Seminar / viva*: The faculty with due approval from the department council shall choose one or more from this category, depending on the nature of subject and the mode of assessment is to be declared at the commencement of the semester. For seminar, topics outside but related to thesyllabus shall be chosen.

iv. *Attendance* :

Attendance	**% of Marks for attendance**
>=90	100
85 to 89	80
80 to 84	60
76 to 79	40
75	20

Practical : The Components of CA for practical courses except Case study I and II are as follows:

	COMPONENTS	**% OF MARKS**
i	Lab Test (Minimum one)	20%
ii	Completion of the list of Lab assignments prescribed by the faculty	20%
iii	Periodical assessment of Lab assignments through execution of programs and viva	40%
iv	Attendance (Mark distribution is same as that of theory)	20%

For Case study I and II :

	COMPONENTS	**% OF MARKS**
i	Periodical viva / short quizzes / short programming assignments to evaluate the basic knowledge/understanding of the tool.	30%
ii	Coding – Logic, Selection of appropriate constructs / features of the Tool, Style etc.	30%
iii	Execution of the case study - output	20%
iv	Viva based on case study	20%

Note :*All the records in respect of Continuous Assessment (CA) must be kept in the department and must be made available for verification by university. The results of the CA shall be displayed on the notice board within 5 working days from the last day of a semester. It should be get signed by the candidates. The marks awarded for various components of the CA shall not be rounded* *off, if it has a decimal part. The total marks of the CA shall be rounded off to the nearest whole number.*

END SEMESTER EVALUATION (ESE):

There shall be double valuation system of answer books. The average of two valuations shall be taken in to account. If there is a variation of more than 10% of the maximum marks, the answer books shall be valued by a third examiner. The final marks to be awarded shall be the average of the nearest two out of three awarded by the examiners. After that there shall be no provision for revaluation

Pattern of questions: Questions shall be set to assess knowledge acquired, standard application of knowledge, application of knowledge in new situations, critical evaluation of knowledge and the ability to synthesize knowledge. Question paper for end semester theory examination shall consist of:

i. Short answer type : 12 questions of which 10 to be answered. 10 x 3 = 30 marks,

ii. Essay type: 5 questions (one either –or question from each module) x 10 marks = 50 marks

End Semester Evaluation in Practical courses shall be conducted and evaluated by two examiners- one internal and one external. Details of scheme of evaluation of ESE practical courses are given along with respective syllabus.

3.4 Project: A project work has to be undertaken by all students. The project can be software development following all or some of the software development lifecycle or an R&D project. The hours allotted for project work may be clustered into a single slot so that students can do their work at a centre or location for a continuous period of time. The Major project work should be carried out in the Department /Institution or in alevel Industry / R & D organization of national repute. Project work shall be carried out under the supervision of a Teacher. If the project is carried out in an Industry / R & D organization outside the campus, then a co-guide shall be selected from the concerned organization. If the project work is of interdisciplinary in nature, a co-guide shall be taken from the other department concerned. Every student should do the Project individually and no grouping is allowed. All the candidates are required to get the approval of their synopsis and the guide before commencement of the project from the Department. A co-guide should be a postgraduate in CS or allied subject or a person of eminence in the area in which student has chosen the project. At the end of the semester the candidate shall submit the Project report (two bound copies and one soft copy) duly approved by the guide and co-guide for End Semester Evaluation. The project report shall be prepared according to the guidelines approved by the University.

Evaluation of Project:

i. A Departmental committee duly constituted by the Head of the Department will review the project periodically.

ii. **Continuous Assessment of project work**: There shall be three internal presentations before the committee (Minimum two members, including the guide). The assessment is based on presentation, interim report and viva voce. The total mark for CA shall be divided among the three presentations in the ratio 20%:30%:50%.Each internal presentation shall be evaluated based on the following components:

Component	% of marks
Understanding of the problem / concepts	25
Adhering to methodology.	20
Quality of presentation and demonstration (Demonstration is optional)	15
Quantum of work / effort	30
Organization and content of mid-term report	10

iii. **End Semester Assessment of Project**: A board of two examiners appointed by the University shall conduct ESE evaluation. The evaluation shall be based on the report, presentation of the work, demonstration of the work (optional) and a detailed viva voce based on the work carried out. A candidate will not be permitted to attend the Project evaluation without duly certified project reports. Also a project will be evaluated only if the candidate attend the ESE presentation and Viva voce on the scheduled date and time. A board shall evaluate a maximum of 10 candidates in a day. The ESE evaluation shall consist of the following components:

Component	% of marks
Understanding of the problem/requirements/ concepts related to the project	15
Adhering to methodology (Software engineering phases or research methodology) and the candidates understanding of the components of methodology	15
Quality of Modeling of the problem and solution/ database design / form design / reports / testing (For research projects - relevance /novelty of the work(s)/ use of data/ proposal of new models /analysis of algorithms/ comparison and analysis of results /findings)	20
Quality of presentation / demonstration	15
Quantum of work / effort - assessed through the content of report, presentation and viva.	25
Organization and content of report	10

iv. A student shall be declared to pass in the Project report course if she/he secures minimum 40 % marks of the aggregate and 40% separately for external.

v. If a candidate fail in the evaluation of Project, he/she has to repeat the project course along with the next batch and undergo both CA and ESE. *Unlike theory/practical courses, the CA mark will not retained.*

vi. There shall be no improvement chance for the marks obtained in the Project course.

3.5 Seminar: Each student shall select a relevant topic, prepare a seminar report and give a presentation (30 to 45 minutes),under the guidance of a faculty member. The evaluation of seminar is 100% internal and components and mode of evaluation shall be formulated by the department council (May include components like content, Presentation, interaction and structure of report).

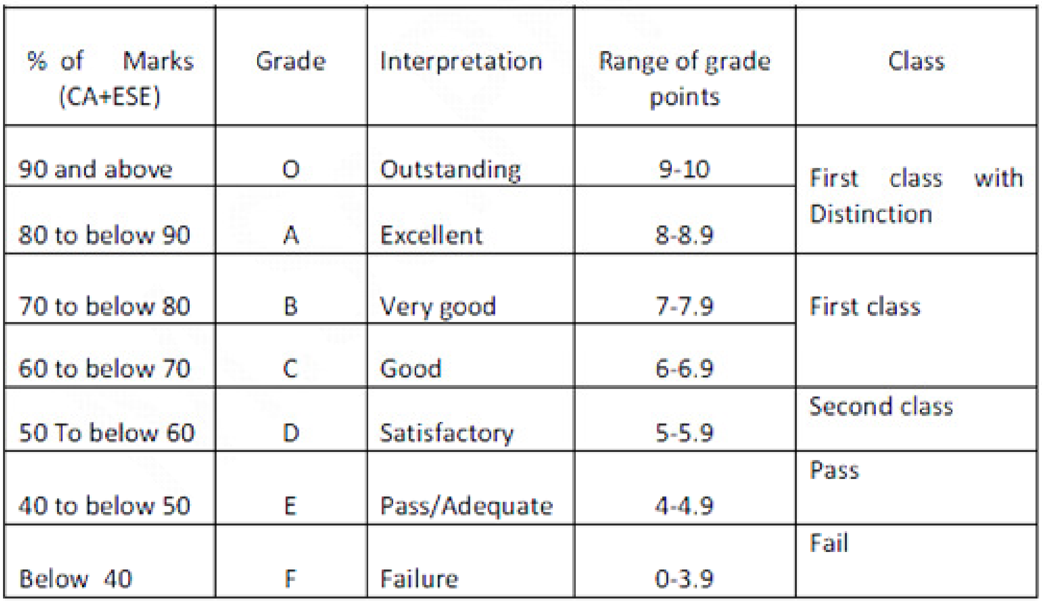
3.6 VIVA VOCE: A general Viva Voce covering all courses in the Programme shall be conducted in the fourth semester. The Viva voce shall be conducted by two external examiners. The Viva voce *shall not be clubbed* with the project evaluation. The details of the mode of conduct and evaluation of Viva Voce shall be decided by the BOE.

4. **GRADING SYSTEM**

Seven Point Indirect Relative grading system:

Evaluation(both internal and external) is carried out using Mark system .The grading on the basis of a total internal and external marks will be indicated for each course and for each semester and for the entire programme.

The guidelines of grading is as follows-



% of Marks (CA+ESE)	Grade	Interpretation	Range of grade points	Class
90 and above	O	Outstanding	9-10	First class with Distinction
80 to below 90	A	Excellent	8-8.9	
70 to below 80	B	Very good	7-7.9	First class
60 to below 70	C	Good	6-6.9	
50 To below 60	D	Satisfactory	5-5.9	Second class
40 to below 50	E	Pass/Adequate	4-4.9	Pass
Below 40	F	Failure	0-3.9	Fail

$$S.G.P.A = \frac{\text{SUM OF CREDIT POINTS OF ALL COURSES IN THE SEMESTER}}{\text{TOTAL CREDITS IN THAT SEMESTER}}$$

CREDIT POINT = GRADE POINT (G) X CREDIT (C)

$$\text{C.G.P.A} = \frac{\text{Sum of credit points of all completed semesters}}{\text{Total credits acquired}}$$

$$\textbf{OGPA} = \frac{\textbf{Sum of } \boldsymbol{credit\ points} \textbf{ obtained in four } \boldsymbol{semesters}}{\boldsymbol{Total\ credits}\ \mathbf{(80)}}$$

PASS REQUIREMENT:

COURSE:

A CANDIDATE SECURING E GRADE WITH 40% OF AGGREGATE MARKS AND 40% SEPARATELY FOR ESE FOR EACH COURSE SHALL BE DECLARED TO HAVE PASSED IN THAT COURSE.

SEMESTER

Those who secure not less than 40 % marks (both ESE and CA put together) for all the courses of a semester shall be declared to have successfully completed the semester.

The marks obtained by the candidates for CA in the first appearance shall be retained (irrespective of pass or fail)

The candidates who fail in theory unit shall reappear for theory unit only, and the marks secured by them in practical unit, if passed in practical, will be retained.

A candidate who fails to secure a minimum for a pass in a course will be permitted to write the same examination along with the next batch.

For the successful completion of a semester, a candidate should pass all courses and secure a minimum SGPA of 4. However a student is permitted to move to the next semester irrespective of his/her SGPA. A student will be permitted to secure a minimum SGPA of 4.00 required for the successful completion of a Semester or to improve his results at ESE of any semester, by reappearing for the ESE of any course of the semester concerned, along with the examinations conducted for the subsequent admission

IMPROVEMENT:

A candidate who secures minimum marks (40 %) for a pass in a course will be permitted to write the same examination along with the next batch if he/she desires to improve his/her performance in ESE. If the candidate fails to appear for the improvement examination after registration, or if there is no change/up gradation in the marks after availing the improvement chance, the marks obtained in the first appearance shall be retained. There shall be no improvement chance for the marks obtained in internal assessment. Improvement of a particular semester can be done only once. The student shall avail the improvement chance in the succeeding year along with the subsequent batch.

There will be no supplementary examinations. For re-appearance/ improvement student can appear along with the next batch.

KANNUR UNIVERSITY

M.Sc. Computer Science with Specialization in Artificial Intelligence

Course Structure and Scheme of Evaluation (From 2020 Admission)

(CBSS- For affiliated Colleges)

The Regulations, Curricula, Syllabus and Scheme of Evaluation of M.Sc. Computer Science with Specialization in Artificial Intelligence are the same as MSc Computer Science programme in affiliated colleges in Kannur University (Implemented with effect from 2014 admission) except elective courses. The syllabus of elective courses are also attached herewith.

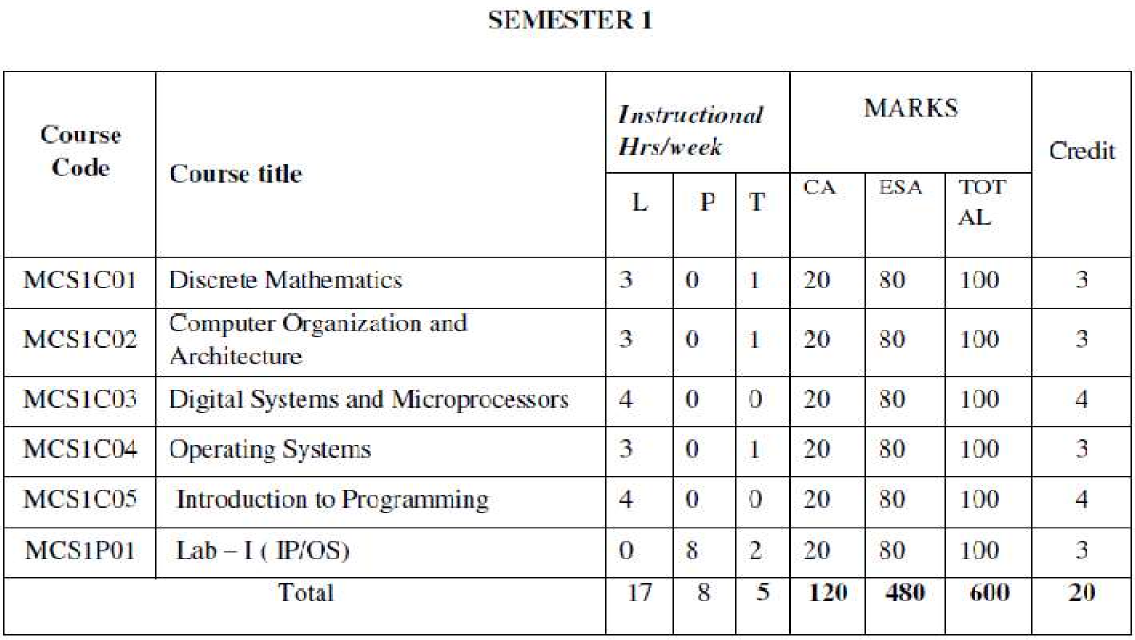
Eligibility for Admission:

1. A pass in BSc Computer Science / BCA/ BE /BTech or equivalent degree.

OR

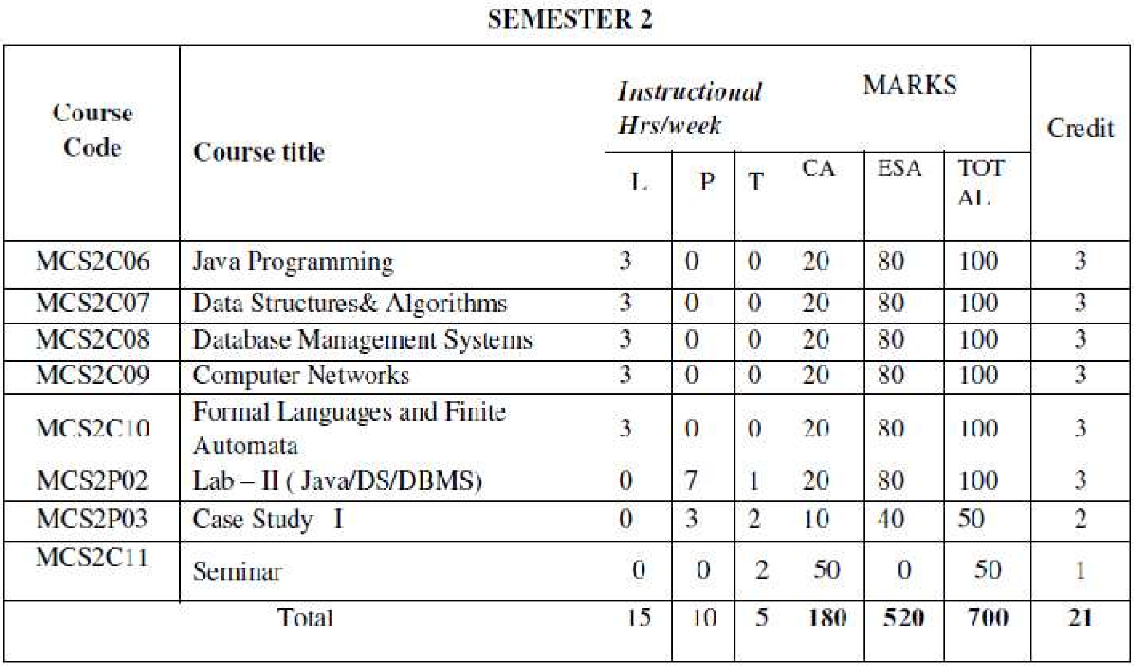
A pass in BSc with Mathematics as Core/Complementary or equivalent degree.

2. Obtained at least 50% marks/grade (45% marks/grade in case of candidates belonging to reserved category) in the qualifying examination.



SEMESTER 1

Course Code	Course title	*Instructional Hrs/week*			MARKS			Credit
		L	P	T	CA	ESA	TOTAL	
MCS1C01	Discrete Mathematics	3	0	1	20	80	100	3
MCS1C02	Computer Organization and Architecture	3	0	1	20	80	100	3
MCS1C03	Digital Systems and Microprocessors	4	0	0	20	80	100	4
MCS1C04	Operating Systems	3	0	1	20	80	100	3
MCS1C05	Introduction to Programming	4	0	0	20	80	100	4
MCS1P01	Lab – I (IP/OS)	0	8	2	20	80	100	3
Total		17	8	5	**120**	**480**	**600**	**20**



SEMESTER 2

Course Code	Course title	*Instructional Hrs/week*			MARKS			Credit
		L	P	T	CA	ESA	TOTAL	
MCS2C06	Java Programming	3	0	0	20	80	100	3
MCS2C07	Data Structures& Algorithms	3	0	0	20	80	100	3
MCS2C08	Database Management Systems	3	0	0	20	80	100	3
MCS2C09	Computer Networks	3	0	0	20	80	100	3
MCS2C10	Formal Languages and Finite Automata	3	0	0	20	80	100	3
MCS2P02	Lab – II (Java/DS/DBMS)	0	7	1	20	80	100	3
MCS2P03	Case Study I	0	3	2	10	40	50	2
MCS2C11	Seminar	0	0	2	50	0	50	1
Total		15	10	5	**180**	**520**	**700**	**21**

Case Study I: Python Programming for Artificial Intelligence and Machine Learning

SEMESTER 3

Course Code	Course Title	Instructional Hrs/week			Marks			Credit
		L	P	T	CA	ES A	Total	
MCS3C12	Computer Graphics	3	0	0	20	80	100	3
MCS3C13	System Programming & Compiler Design	3	0	0	20	80	100	3
MCS3C14	System Administration and Network Programming	3	0	0	20	80	100	3
MCS3C15	Software Engineering	3	0	0	20	80	100	3
MCS3C16	Research Methodology	1	0	1	50	0	50	1
MCS3C17	Artificial Intelligence	3	0	0	20	80	100	3
MCS3P04	Lab – III (CG /NP&A/SP&CD)	0	6	2	20	80	100	3
MCS3P05	Case study II *	0	3	2	10	40	50	2
	Total	16	9	5	180	520	700	21

* Case Study-II : Apply of AI methods to real word applications

SEMESTER 4

Course Code	Course Title	Instructional Hrs/week			Marks			Credit
		L	P	T	CA	ES A	Total	
MCS4C18	Soft Computing Techniques	3	0	0	20	80	100	3
MCS4C19	Machine Learning	3	0	0	20	80	100	3
MCS4C20	Deep Learning	3	0	0	20	80	100	3
MCS4Pr04	Project #		16	5	20	80	100	7
MCS4C21	General Viva Voce	-	-	-	0	100	100	2
	Total	9	16	5	80	420	500	18

# Project based on Artificial Intelligence/Machine Learning / Deep Learning

SYLLABUS
CORE COURSES
(Theory)

MCS1C01 DISCRETE MATHEMATICS

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Propositional logic – Propositions, truth tables, converse, contra positive and inverse, compound statements and their truth tables, translating natural language sentences to logical statements, tautology, contradiction, logical equivalence, De Morgan's laws, normal forms.

Predicate logic – predicates, universal and existential quantifiers, binding variables, translating natural language sentences to logical statements.

Unit 2

Sets, representation of sets, set operations, Cartesian product, using set notation with quantifiers, truth sets of quantifiers, computer representation of sets.Functions – one-to-one and onto functions, inverse functions and compositions of functions.

Unit 3

Relations – properties, functions as relations, relations on a set, combining relations, n-ary relations and their applications, representing relations, closures of relations, Warshall's algorithm, equivalence relations, equivalence classes and partitions.

Unit 4

Basics of counting, basic counting principles, the inclusion-exclusion principle, the pigeonhole principle, the generalized pigeonhole principle, permutations and combinations, with and without repetitions.Generating permutations and combinations.Recurrence relations, modeling with recurring relations.

Unit 5

Graphs – definition, different types of graphs, graph models, basic terminology, representing graphs, isomorphism, connectivity, Euler and Hamilton paths, shortest path problem and Dijkstra's algorithm.

Trees - basic terminology, properties (no proofs), spanning trees, depth-first and breadth-first searches.

Reference books:

1. Kenneth H. Rosen, Discrete Mathematics and Applications, TMH 2003

2. J.P. Tremblay and R Manohar Discrete Mathematical Structure with Applications to Computer Science, TMH 2001.

3. John Truss, Discrete Mathematics for Computer Scientists, Pearson Edn 2002

4. Sengadir, Discrete Mathematics, Pearson, 2009

MCS1C02 COMPUTER ORGANIZATION AND ARCHITECTURE

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Basic structure :Basic operational concepts. Number representation and arithmetic operations.Character representations.Performance.

Instruction set Architecture:Memory locations and addresses, memory operations, instructions and instruction sequencing, addressing modes. Assembly language, stacks, subroutines, RISC vs CISC.

Unit 2

Basic I/O: Accessing I/O devices (device interface, program controlled I/O), Interrupts (enabling and disabling, handling multiple interrupts, controlling I/O device behavior, Processor control registers, exceptions).

I/O organization: Bus structure, bus operation, arbitration, Interface circuits, interconnection standards (USB, PCI, Firewire, SCSI, SATA).

Unit 3

Basic Processing Unit : Fundamental concepts, Instruction execution, Hardware components, Instruction fetch and execution steps, control signals, Hardwired control, CICS style processors (3-bus organization, microprogrammed control).

Arithmetic - multiplication of unsigned numbers (array and sequential multipliers), multiplication of signed numbers (Booth algorithm), Fast multiplication (bit pair recoding), Floating point numbers and operations.

Unit 4

Memory system : Basic concepts, Semiconductor RAMS, ROMs, DMA, Memory hierarchy, Cache memory, performance requirements, virtual memory, memory management requirements, secondary storage devices.

Unit 5

Pipelining: basic concepts, pipeline organization, issues, data dependencies, memory delays, branch delays, performance evaluation, superscalar operations.

Parallel processing: Hardware multithreading, Vector processing, Shared memory multiprocessors, message passing multi-computers.

Text book:

1. Hamacher, Vranesic, Zaky, Manjikian, Computer Organization and Embedded Systems, 6thedn, Tata McGraw Hill.

Reference books:

1. William Stallings, Computer Organization & Architecture – Designing for Performance, 9th Edn, Pearson
2. John P. Hayes, Computer Architecture and Organization, Third Edn, Tata McGraw Hill.
3. M. Morris Mano, Computer System Architecture, PHI 2003

MCS1C03 DIGITAL SYSTEMS & MICROPROCESSORS

Contact Hours/ week: 4 **Credit: 4**

Unit 1.

Gates, Boolean algebra & Laws, Combinational Circuits : SOP, POS, K-Map Simplification (up to 6 variables), Tabular method, Decoders, Multiplexer, De-multiplexer, Encoder, Adders: Half Adder, Full Adder, Cascading Full-Adders, Look-Ahead Carry, Logic Families: RTL, DTL, I^2L, TTL, ECL, MOS, FETs, MOSFETs, CMOS..

Unit 2

Sequential circuits: Flip-flops: SR, JK, D, Master-Slave, Edge-Triggered, T flip- flops Registers: Registers with parallel load, Shift Registers, Bidirectional Shift Registers with parallel load, Tristate Registers, Counters: Design, Simple Counters(Divide by 2,4 and 8, Johnson Counter, Ring Counter), Ripple Counters, Synchronous Counters.

Unit 3

Microprocessors: Architecture of 8085, Block Diagram and pin outs, Instruction set, Addressing modes, Subroutines, Interrupts, Peripheral Interfacing.8255A Programmable peripheral interfacing: 8254 programmable interval timer, 8237 DMA Controller, 8279 Keyboard/Display Controller.

Unit 4

Advanced Microprocessors: Architecture of 8086, Additional features of 8086: Pin diagrams, Timing Diagrams, Addressing Modes, Memory organization – segment-offset addressing, , Min-Max mode, Stack structure, Interrupts.

Unit 5

Special Features of advanced processors 386,486 and Pentium: Memory System, I/O System, Timing, Registers, Memory Management. 386: Moving to protected mode, Virtual mode, Memory paging mechanism. Pentium: Extensions- Introduction.

Reference Books

1. John . M. Yarbrough , Digital Logic Applications and Design,Thomson -2002 .
2. M. Moris Mano, Digital Design – PHI 2001
3. R. Gaonkar, Microprocessor Architecture and Programming.TMH-2002.
4. Bary B. Brey, The Intel microprocessors , PHI 2003

MCS1C04 OPERATING SYSTEMS

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Introduction – Mainframe systems, Desktop systems, Multiprocessor systems, Distributed systems, Clustered systems, Real time systems, Hand held systems, Computing environments. **Computer System structures** – Computer system operation, I / O Structure, Storage structure, Storage hierarchy, Network structures. **Operating system structures** - System components, Operating systems services, System calls, System programs, System structure, Virtual machine, System design and implementation.

Unit 2

Processes – Process concepts, Process scheduling, Operations on Process, Cooperating Process, Inter Process communication in Client/ Server system. **Threads**- Multi threading models, Threading issues, Pthreads, Linux and Java Threads. **CPU Scheduling** – Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor Scheduling, Real time Scheduling, Algorithm evaluation, Process Scheduling models. **Process Synchronization** – Critical section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, Critical region, Monitors, OS Synchronization, Atomic transaction.**Deadlocks** – System models, Deadlocks characterization, Method for handling Deadlocks, Deadlock prevention, Deadlock avoidances, Deadlock detection, Recovery from Deadlocks.

Unit 3

Memory management- swapping, Contiguous memory allocation, Paging Segmentation, Segmentation with paging.**Virtual memory**- Demand paging, processes creation, page replacement, allocation of frames, thrashing. **File system interface and Implementation**- File concepts, access methods, directory structure, File system mounting, File sharing, Protection, File system structure, File system implementation, Directory implementation, Allocation methods, Free space managements, Efficiency and performance, Recovery , Log- structured file system, NFS.

Unit 4

I / O Systems - I / O hardware, Application I/O interface, Kernel I / O subsystem, Transforming I / O to hardware operations, STREAMS, Performances. **Mass storage structure** - Disk structure, Disk scheduling, Disk management, Swap space managements, RAID structure, Disk attachments, Stable storage implementation, Tertiary storage structure.

Unit 5

Distributed Systems – Motivation, Types of Distributed Operating systems.**Distributed file systems** – Background, Naming and transparency, Remote file access, Stateful versus stateless service, File replication. **Protection**- Goals and principlesof protection, Domain of protection, Access matrix, Access control, Revocation of access rights, Capability based systems (Hydra), Language based protection(protection in java). **Security**- The security problem, Program threats, System and network threats.

Text Book:

1. Silberschatz, A., Galvin, P.B. & Gagne, G. "Operating System Concepts", 6th Ed. Wiley-India.

References:

1. Dhamdhere, D. M. "Operating Systems", 2nd Ed. The McGraw - Hill Companies.
2. Kochan, S, G., Wood, P., "Unix shell programming", 3rd ed. Pearson Education, 2003
3. Ditel, Deital and Choffness, Operating Systems, Pearson, 3rdEdn

MCS1C05 INTRODUCTION TO PROGRAMMING

Contact Hours/ week :4 **Credit : 4**

Unit 1

Introduction to „ C" programming –fundamentals – structure of a C program – compilation and linking processes – Constants, Variables – Data Types – Expressions using operators in C – Managing Input and Output operations – Decision Making and Branching – Looping.

Arrays – Initialization – Declaration – One dimensional and Two dimensional arrays. String- String operations – String Arrays.Simple programs- sorting- searching – matrix operations.

Unit 2

Function – definition of function – Declaration of function – Pass by value – Pass by reference – Recursion. Pointers - Definition – Initialization – Pointers arithmetic – Pointers and arrays- Structure definition – Structure declaration – Structure within a structure - Union - Example programs.Storage classes, Pre-processor directives.

Unit 3

Introduction to OOP – overview of C++,Class, Object, inline functions, constructors, destructors, scope resolution operator, friend functions, friend classes, static members, *this* pointer, references, dynamic memory allocation.

Unit 4

Function overloading, overloading constructors, pointers to functions Operator overloading.
Inheritance, types of inheritance, protected members, virtual base class, polymorphism, virtual functions, pure virtual functions.

Unit5

Streams, formatting I /O with class functions and manipulators, overloading << and >> , File I/ O , name spaces, conversion functions, array based I /O, Standard Template Library (STL), Class templates and generic classes, function templates and generic functions.

References:

1. Kernighan,B.W and Ritchie,D.M, "The C Programming language", 2ndEdn, Pearson Education, 2006
2. Balagurusamy, Programming in ANSI C, 5thedn, TMH.
3. Byron Gorrfried, Programming with C, 3rdEdn, Schaum's outline.
4. Schildt, C++ - The complete Reference, 4thedn, McGraw Hill.
5. Somashekara, Guru, Nagendrasamy, Majunath, object Oriented Programming with C++, 2ndedn PHI
6. BjarneStroustrup - The C++ Programming language, Addison Wesley , 3rd Ed.

MCS2C06 JAVA PROGRAMMING

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Object oriented programming, basic concepts of OOP; Introduction to Java programming, features of Java: - Bytecode, Java Virtual Machine (JVM), Java Applets and Applications, Java file name and directory structure; Packages of Java API. Data Types, Variables, and Arrays, Type Conversion and Casting; Operators; Control Statements.

Unit 2

Class, Class Fundamentals, Declaring Objects, Constructors, access specifier, static, Nested and Inner Classes, Command-Line Arguments, this Keyword; Garbage Collection.String handling. Collection class.

Inheritance, method overloading, Method Overriding, Dynamic Method Dispatch, Abstract Classes.

Unit 3

Packages, Importing Packages; Interface: Defining an Interface, Implementing Interfaces; Exception Handling: try, catch, throw, throws, and finally, Java's Built-in Exceptions; Thread, Synchronization, Messaging, Runnable interface, Inter thread communication, Deadlock, Suspending, Resuming and stopping threads, Multithreading. I/O streams, File streams.

Unit 4

Applets: Applet lifecycle, working with Applets, The HTML APPLET tag. Working with Graphics. Abstract Window Toolkit (AWT): AWT Classes, Window Fundamentals, Component, Container, Panel, Window, Frame. working with Frame Windows, AWT Controls, Layout Managers, and Menus.

Unit 5

Event Handling: Events, Event Sources, Event Classes, Event Listener Interfaces, Adapter Classes.

Java database connectivity:-jdbc architecture- drivers- database connections- statements-resultsets- transactions-metadata-stored procedures-error handling- blobs and clobs.

Reference books:

1. Herbert Schildt, The complete reference Java2 ,7thed, Mc, Graw Hill.
2. David Flanagan, Java in a Nutshell A desktop quick Reference, 2 Edition, OReilly&Associates, Inc
3. Java programming, Rajkumar, pearson, 2013
4. Java Programming, HarimohanPandey, Pearson, 2012
5. Core Java for beginners, sha and sha, ShroffPubl and dist, 2010
6. Object Oriented Programming through Java, Rasdhakrishnan, University Press, 2007
7. Java for Programmers, 2ndEdn, Deital and Deital, Person

MCS2C07 DATA STRUCTURES & ALGORITHMS

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Abstract Data Types (ADT**),** Algorithm analysis, Asymptotic notations.

Arrays – representation. Polynomials with arrays – operations – addition and evaluation.Sparse matrix representation with arrays – operations- transpose and addition.

Linked list – Singly linked list (SLL) – basic operations (create list, add/delete nodes, traverse/print, search SLL, concatenate, merge two sorted SLLs, recursive function for reversing a SLL). Circular SLL – operations (add/delete nodes, print, concatenate, search). LL with header/trailer nodes. Doubly Linked List – basic operations (create list, add/delete nodes, traverse/print). Polynomials with SLL – addition and evaluation.

Unit2

Stack – array and Linked List implementation – applications – infix to postfix conversion – evaluation of postfix. Queue – array and Linked implementation – circular array Queue – Priority Queue – implementation with array and LL. Application of queues.

Non-linear data structures – tree and binary tree– basic definitions and properties –function to create binary tree - traversal – recursive and non-recursive, Print/traverse data level by level, count number of nodes.

Unit3

Threaded binary tree(TBT) – inorder threaded BT and function for inorder traversal of Inorder TBT. Binary search tree – create - add/delete nodes – search. Applications of trees.

AVL trees –B-Trees – Red-black trees (Basic ideas only).

Hashing - Hashing functions - Collision Resolution Techniques - Separate chaining - Open addressing – Multiple hashing.

Unit 4

Graph - Definitions – Representation of graph - Graph Traversals - Depth-first traversal – breadth-first traversal - applications of graphs – shortest-path algorithm – Dijkstra's algorithm **-**minimum spanning tree – Prim's and Kruskal's algorithms.

Unit 5

Sorting – Insertion,Quick and Heap.

Algorithms - Divide and Conquer – Merge Sort – Binary Search - Greedy Algorithms – Knapsack Problem – Dynamic Programming – Warshal's Algorithm for Finding Transitive Closure – Backtracking – Sum of Subset Problem.

Reference Books:

1. Horowitz, Sahni and Mehta, Fundamentals of Data Structures in C++, 2ndEdn, University Press
2. Horowitz, Sahni, Rajasekaran, Fundamentals of Algorithms, 2ndEdn, University Press
3. M. A. Weis, Data Structures and Algorithm Analysis in C++, Pearson Education Asia, 2013
4. Langsam, Augenstein and Tenenbaum, Data Structures Using C and C++, 2ndedn, PHI.
5. Anany Leviton, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson Education.
6. Aho, Hopcroft and Ullman, Data Structures and Algorithms, Pearson Education.

MCS2C08 DATABASE MANAGEMENT SYSTEMS

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Database concepts, Relational database : Introduction to Relational model , relational algebra, views, tuple relational calculus, domain relational calculus, SQL- basic structure, set operations, sub queries, joint relation, DDL, DML, embedded SQL, QBE. Formal relational query language.

Unit 2

Database design : ER model basic concepts, constraints, Keys, ER diagram, Reduction of ER schema, UML, design of an ER database schema. Relational database design - 1st, 2nd, 3rd, 4th, BCNF, 5th Normal forms. Integrity and security, domain constraints, referential integrity, assertion, triggers, authorization in SQL, relational database design

Unit 3

Data storage and querying – storage and file structures, Indexing and hashing, basic concepts, static hashing, dynamic hashing, multiple key accesses, Query processing- Query optimization Transaction Management-Transaction concepts, transaction definition in SQL. Concurrency control, Recovery systems, deadlock handling.

Unit 4

Database system Architecture, Parallel databases, distributed databases, Data warehousing and mining- object based databases

Unit 5

Case study : PostgreSQL – data type – tables – psql – operations on tables – sub queries – views - operators & functions –indices – arrays – transactions and cursors, PostgreSQL Administration – authentication and Encryption – Database management – User and group management – PostgreSQL programming – Pl/pgSQL.

Reference Books

1. Silbersehatz, Korth and Sudarshan, Database system concepts, 6th edition MGH 2011
2. Ramakrishnan and Gehrke, Database Management Systems, 3rd Edn, McGraw Hill, 2003
3. A Leon & M Leon, Database Management Systems , Leon Vikas – 2003.
4. Elmasri and Navathe, Fundementals of Database systems, 5thEdition ,Pearson 2009
5. O'Reilly, Practical PostgreSQLShroffPublishers(SPD) 2002

MCS2C09 COMPUTER NETWORKS

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Introduction, network hardware, software, Reference Model,Internet, ATM, Physical Layer, Transmission Media, Wireless transmission, Switching – circuit switching, packet switching, message switching and hybrid switching - Communication Satellites.

Unit 2

Data Link Layer design issues, Error detection & correction, Elementary data link protocols, Sliding Window protocols, Data Link Layer in the Internet.

Unit 3

Medium access layer, Channel allocation problem, Multiple access protocols, Ethernet, Wireless LAN, Bluetooth.

Unit 4

Network Layer, Design Issues, Routing Algorithms, Congestion Control Algorithm, Internetworking, Internet Protocol, IP Address, Internet Control Protocol.

Unit5

Transport Layer, Design Issues, Connection Management – addressing, establishing and releasing a connection, Simple Transport Protocol, Internet Transport Protocol, E-mail, Network Security, Cryptography.

Text book

1. Andrews S. Tanenbaum. "Computer Networks", 4th Edition, Prentice Hall of India, 2006.

References Books

1. Behrouz A Forouzan. "Data Communications and Networking", 4th Edition, McGraw Hill, India, 2011.
2. William Stallings. "Data and Computer communications", 7th Edition, Prentice Hall of India, 2004.
3. Kruse and Ross, Computer Networking, , 5thedn, Pearson

MCS2C10 FORMAL LANGUAGES AND FINITE AUTOMATA

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Introduction to the Theory of computation and Finite Automata: Mathematical preliminaries and notation, Proof techniques, Three basic concepts: languages, grammar & automata. Some applications.

Finite automata: Deterministic Finite Acceptors, Nondeterministic Finite Acceptors, Equivalence of deterministic and nondeterministic finite acceptors, Reduction of the number of states in finite automata.

Unit 2

Regular Languages and Regular grammars :Regular expressions, connection between regular expressions and regular languages , regular grammars

Properties of Regular Languages:closure properties of regular languages, identifying non regular language.

Context-free grammars & languages Context-free grammars, parsing and ambiguity.

Unit 3

Simplification of Context free Grammars and Normal forms : methods of transforming grammars, two important normal forms.

Pushdown automata for context-free languages Non deterministic pushdown automata, PDA and context-free languages, deterministic pushdown automata and deterministic context-free languages.

Unit 4

Properties of Context-Free Languages: pumping lemmas for context free languages and linear languages, closure properties for context-free languages.

Turing machine Standard Turing machine, combining Turing machines for complicated tasks, Turing's thesis

Unit 5

Other models of Turingmachine : Minor variations on the Turing machine theme, Turing machine with complex storage, nondeterministic Turing machine, a universal Turing machine, Linear bounded automata.

Limits of Algorithmic computation: Problems that cannot be solved by Turing machines, Undecidable Problems for Recursively enumerable Languages, The Post Correspondence problem.

Text Book :

1. An introduction to Formal Languages and Automata, Peter Linz, 4thedn, Narosa publishing House.

Reference Books

1. John C Martin, Introduction to Languages and the Theory of Automata, McGraw Hill 1997
2. Mishra &Chandrasekharan, Theory of Computer Science : Automata, Languages and Computation, 3rd edn, PHI
3. Hopcroft, Motwani and Ullman, Introduction to automata theory, Languages and Computation, 3rdEdn,., Pearson

MCS3C12 COMPUTER GRAPHICS

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Overview of Graphics systems: Video display devices, Raster scan systems, Graphic workstations and viewing systems, Input devices, Graphics software, introduction to OpenGL.

Graphics Output Primitives:Coordinate reference frames, Line drawing algorithms (DDA and Bresenham's), OpenGL curve functions, Circle generating algorithms (Midpoint circle), Pixel addressing and Object geometry, fill area primitives, Polygon fill areas.

Unit 2

Attributes of graphics primitives : Color and Gray scale, point attributes, Line attributes, Fill-Area attributes, General Scan-line polygon fill algorithm, Scan-Line fill of convex-polygons, Boundary fill and flood fill algorithms, Antialiasing.

Two-dimensional viewing : 2D viewing pipeline, Clipping window, normalization and viewport transformation, Clipping algorithms, point clipping, line clipping (Cohen-Sutherland, Nichol-Lee-Nichol), Polygon Fill-area clipping (Sutherland – Hodgeman).

Unit 3

Geometric Transformations: Basic 2D transformation, Matrix representation and Homogeneous coordinates, Inverse transformations, 2D composite transformations, Reflection and shear, Raster methods for geometric transformations, Transformations between 2D coordinate systems.3D Geometric transformations, 3D translation, 3D rotation (coordinate axis rotation, General 3-d rotation, Quaternion methods for 3D rotation), 3D scaling, 3D composite transformations, transformations between 3D coordinate systems.

Unit 4

Three-dimensional viewing : Overview of 3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformations, orthogonal projections (axonometric and isometric, orthogonal projection coordinates, clipping window and orthogonal projection view volume, Normalization transformation), Oblique parallel projections (Cavalier and cabinet projections, Clipping window and Oblique parallel-projection view volume, Oblique parallel projection transformation matrix, normalization transformation), Perspective projections (transformation coordinates, perspective-projection equations, vanishing points, view volume, transformation matrix, symmetric and oblique perspective-projection frustum, Normalized perspective-projection transformation coordinates), 3D clipping algorithms (region codes, point and line clipping, polygon clipping)..

Unit5

3D Object representation : Quadric surfaces, superquadrics, blobby objects, spline representations.

Visible surface detection methods : Classification, Back-face detection, depth-Buffer method, A-buffer method. Wireframe visibility methods.

Illumination models and surface rendering methods : Light sources, Surface lighting effects, Basic illumination models (Ambient light, Diffuse reflection, Specular reflection and the Phong model), polygon rendering methods (constant intensity surface rendering, Gouraud surface rendering, Phong surface rendering), Ray tracing methods – basic Ray-tracing algorithm.

Text Book :

1. Hearn and Baker, Computer Graphics with OpenGL, 3rdedn, Pearson.

Reference Books:

1. Hill Jr. and Kelly, Computer Graphics using OpenGL, 3rdEdn, Pearson
2. Shreiner, Sellers, Kessenich, Licea-Kane, OpenGL programming guide, 8thedn, Pearson.
3. Foley, Van Dam, Feiner, Hughes, Computer Graphics- Principles and practice, Second Edition in C, Pearson Education.

MCS3C13 SYSTEMS PROGRAMMING & COMPILER DESIGN

Contact Hours/ week : 3 **Credit : 3**

Unit l

Assemblers: Elements of Assembly Language Programming, Overview of Assembly Process, Design of Two pass Assembler, Macros and Macro Processors, Macro definition, call and expansion, Nested Macro calls, Advanced Macro facilities, Design of Macro preprocessor.

Unit 2

Linkers: Linking and Relocation concepts, Design of linkers, Self relocating programs, Linking for over-lays, Loaders. Introduction to compilers: Different Phases. Lexical Analysis:role of the lexical analyzer, input buffering, specification of tokens, Recognition of tokens, lexical Analyzer generators, Lex.

Unit 3

Syntax Analysis: role of the parser Context free grammar, writing a grammar, Top down parsing, Recursive descent parsing, Predictive parsing. Bottom Up Parsing, Shift Reduce parsing, Operator precedence parsing, LR parsers (SLR, Canonical and LALR). Parser generators, Yacc.

Unit 4

Syntax-directed translation – Syntax-directed definitions: S-attributed definition, L-attributed definition. Top-down and bottom-up translation, Type checking, Type systems, Specification of a type checker. Run time Environment:source language issues, storage organization Storage organization schemes, Activation records. Storage allocation strategies, Access to non-local names. Parameter passing mechanisms.Symbol tables.

Unit 5

Intermediate code generation, intermediate languages, declaration and assignment statements. Code generation: Issues, target machine, run time storage management, Runtime storage allocation, basic blocks and flow graphs. Code optimization: Principal sources of optimization.

Text books:

1. D.M. Dhamdhree, "Systems Programming and Operating Systems", TMH, 2003.
2. A.V. Aho, R. Semi, J.D. Ullman, "Compilers - Principles, techniques and tools", Pearson Education, 2003

Reference books:

1. A.V. Aho and J.D. Ullman, " Principles of Compiler Design", Narosa, 2002
2. Kenneth.C.Louden, Compiler Construction:Principles And Practice, Thomson Learning, India
3. Dave and Dave, Compilers – principles and practice, pearson, 2012
4. Lex and Yac, o'Reilly, 2ndEdn
5. Appel, Modern Compiler Implementation in C, Cambridge , 2012

MCS3C14 SYSTEM ADMINISTRATION AND NETWORK PROGRAMMING

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Introduction:Important parts of kernel; Major services in a UNIX system: init, login from terminals, syslog, periodic command execution crown and at;***Boot process**:* The LILO boot process: LILO parameters, /etc/lilo.conf; The GRUB boot process; The /boot directory and files; initrd file and mkinitrd; Run levels: */etc/inittab, start-up script /etc/rc.d/rc.sysinit;* ***System Configuration**:* The /etc/sysconfig/… files,*kernel modules;* kernel daemon; /etc/conf. modules and module parameters; /lib/modules/… directory structure and contents.

Unit 2

File system configuration: file system types, /etc/fstabloyout and meaning; Basic user environment: /etc/skel/… and home directories, Window manager configuration file locations; ***System Security*:** Host security: tcp_wrappers and /etc/hosts.allow and /etc/hosts. deny, /etc/security, shadow password, file permissions, users groups and umask; Adding and deleting users;***System maintenance*:** Syslogd, klogd and /etc/syslog.conf; Using a remote syslog; The system crontab, dailyscript, tmpwatch and logrotate; Using and managing the system log files; Basic system backup and restore operations; Emergency rescue operations.

Unit 3

TCP / IP Network Configuration: Introduction to TCP / IP network, Protocols, IP address, Hostname, Configuring a Host : setting the host name, assigning IP address, broad cast, net mask and name server address, Editing Host and network files, Interface Configuration: loop back interface, Ethernet interface, The SLIP and PPP interface, Configuring Gateway, Routing through gateway, Network commands: ifconfig, netstat, route. Network applications Configuration: File Transfer Protocol (FTP) and Trivial File Transfer Protocol (TFTP), Network File Systems (NFS) . Network Information System(NIS),Hyper Text Transfer Protocol (HTTP) and Web server, Server Message Block (SMB) Protocol and Samba server, Dynamic Host configuration Protocol (DHCP) Firewalls, Remote booting.

Unit 4

Domain Name Services (DNS) and Mail services: working of DNS, Host name Resolution Name lookup with DNS, Reverse Lookup, Domain Name Servers and Zones, DNS database: SOA, NS, MX, A and PTR records, Secondary and primary DNS, Zone change notification, root servers, internet root domains, configuring DNS, Using nslookup. Simple Mail Transfer Protocol (SMTP), Post office Protocol(POP) Multipurpose Internet Mail Extension (MIME), SMTP and POP3 command, Mail routing, Configuring A mail server.

Unit 5

Inter Process Communication programming : Create a process- fork() system call, Parent and Child Process, Process ID, User and Group ID Half Duplex Unix Pipes, Named Pipes, (First In First Out) , Streams and messages, System V IPC :Message Queues, Semaphores, Shared memory, Sample programs for IPC that uses Pipes, FIFO;Socket Programming: Overview, socket address, Elementary Socket System Calls: socket, socket pair, bind, connect, listen,accept, send, sendto, recv, recvfrom, close, Byte ordering routines, Byte Operations, Address conversion routines, Simple client Programs that uses some reserved ports, Simple Client / Server Program using unreserved ports.

Reference Books

1. Evi Nemeth ., et al, Linux Administration Hand Book , PHI 2003
2. Nicholas Wells, Linux Installation and Administration, Thomson Vikas 2000.
3. Olaf Kirch& Terry Dawson, Linux Network Administrators Guide, O'relly, 2003
4. Hunt, Linux DNS server Administration, BPB Publication, 2003
5. W Richard Stevens, Unix Network Programming, PHI, 2002

MCS3C15 SOFTWARE ENGINEERING

Contact Hours/ week : 3 **Credit : 3**

Unit 1

Software and Software Engineering: Nature of software and web apps, The software process, Software Engineering practice, Software myths.

Process Models: A generic process model, Prescriptive process model, Specialized process models, The unified process, Personal and team process models, Process technology, Product and process.

Agile Development: Agility- Agility and cost of change, Agile process, Extreme programming, Other agile process models.

Unit 2

Project Management Concepts: The management spectrum, People, Product, Process, Project, W^5HH principle.

Product Metrics:A framework for product metrics, Metrics for the requirements model, Metrics for the design model, Design metrics for Web apps, Metrics for Source code, Metrics for Testing, Metrics for maintenance.

Process and project Metrics:Metrics in the process and project domains, Software measurements, Metrics for software quality.

Estimation for Software Projects:Observations on estimation, The project planning process, Software scope and feasibility, Resources, Software project estimation, Decomposition techniques, Empirical estimation models, Specialized estimation techniques.

Project scheduling: Basic concepts, Project scheduling, Defining a task set for software project, Scheduling, Earned value analysis.

Risk Management: Reactive Vs proactive risk strategies, Software risks, Risk Identification, Risk projection, Risk refinement, Risk mitigation, Monitoring, Management, The RMMM plan.

Unit 3

Quality Concepts: Software quality, Software quality dilemma, Achieving software quality.

Review Techniques: Cost impact of software defects, Defect amplification and removal, Review metrics and their use, Informal reviews, Formal technical reviews.

Software Quality Assurance: Elements of software quality assurance, SQA tasks, Goals and metrics, Formal approaches to SQA, Statistical quality assurance, Software reliability, The SQA plan.

Software Configuration Management: Software configuration management, The SCM process, Configuration management for web apps

Principles That Guide Practice: Software engineering knowledge, Core principles, Principles that guide each framework activity.

Understanding Requirements: Requirements engineering, Establishing the ground work, Eliciting requirements, Building requirements model, Negotiating requirements, Validating requirements.

Unit 4

Requirements Modeling: Flow, Behavior and Web Apps, Requirements modeling strategies, Flow oriented modeling, Creating a behavioral model, Requirements modeling for web apps.

Design Concepts: The design process, Design concepts, the design model. Software architecture, Architectural Design, Architectural mapping using dataflows.

Component Level Design: Cohesion, Coupling, Component level design for web apps, Component based development.

Use Interface Design: The golden rules, Use interface analysis and design, Interface analysis, Interface design steps, Web apps interface design, Design evaluation.

Web apps Design: Web apps design quality, Design goals, Design pyramid for web apps, Web apps interface design, Aesthetic design, Content design, Architecture design, Navigation design, Component level design.

Unit 5

Software Testing Strategies A strategic approach to software testing, strategic issues, Test strategies for conventional software, test strategies for web apps, Validation testing, system testing, The art of debugging.

Testing Conventional Applications: Software testing fundamentals, white box testing, Basis path testing, Control structure testing, Black box testing.

Testing Web Applications: Testing concepts for web apps, content testing, User interface testing, Component level testing, Navigation testing, Configuration, Performance and security testing.

Maintenance and Re Engineering: Software maintenance, Reengineering, Software reengineering, Reverse engineering, Restructuring, Forward engineering.

Text book :

1. Roger S. Pressman.Software Engineering – A practitioner's Approach, 7thEdition.,McGraw Hill,2010.

Reference Books:

2. Ian Somerville., Software Engineering., 9th Edition, Pearson , 2012.
3. Richard Fairley. , Software Engineering Concepts , TMH, 1997.
4. PankajJalote., Software Engineering - A precise Approach,Wiley India, 2011
5. Ammann and Offcut, Introduction to Software Testing, Cambridge University Press, 2008

MCS3C16 RESEARCH METHODOLOGY

Contact Hours/ week :1+1 **Credit : 1**

Unit 1

Introduction to Research Methodology : Meaning of Research, Objectives of Research, Motivations in Research, Types of Research, Research Approaches, Significance of Research, Research Process, Creativity and innovation, Thinking skills, Critical Thinking, Productive Thinking, Experimental Skills; Problem Solving Strategies, Logical thinking, Inductive and Deductive logic. Criteria of a good research, Defining the Research Problem: Selecting the Problem, Motivation behind the Problem definition, Techniques in defining the problem.

Unit 2

Research Ethics, Plagiarism, Research Formulation: Selecting the problem, Importance of literature review in selecting a problem, Literature review, primary and secondary sources, reviews, treatise, monographs, web as a source, searching the web.

Unit 3

Critical literature review, Identifying gap areas from literature review, Development of working hypothesis. Research Design: Planning and designing experiments, Critical Analysis

Unit 4

Structure and Components of Research Report, Data Presentation, Types of Report, Layout of Research Report, Mechanism of writing a research Thesis, Formats of a research paper (Science/ Engineering/ Technology research papers), IMRAD format, IEEE/ACM Professional Societies paper formats, Reference Citing Styles.

Unit 5

Publication Process: Peer review process, Open Access publications, other emerging trends in research communications, Shodhganga, Advanced academic search skills in Internet, Google Scholar, Scopus, Impact Factor, h-Index, g- index, Copyrights and Patents, IPR Laws.

REFERENCES

1. Kothari, C.R., “Research Methodology: Methods and Techniques”, New Age Publisher, 2006.
2. Michael P. Marder, “Research Methods for Science”, Cambridge University Press, 1st Ed., 2011.
3. Donald H. McBurney, “Research Methods”, 5th Edition, Thomson Learning, 2006.
4. Ranjit Kumar, “Research Methodology: A Step-by-Step Guide for Beginners”, SAGE Publications, 3rd Ed., 2010.

SYLLABUS
AI COURSES

MCS3C17: ARTIFICIAL INTELLIGENCE

UNIT - 1

Introduction - Overview of AI applications. Introduction to representation and search. The Propositional calculus, Predicate Calculus, Using Inference Rules to produce Predicate Calculus expressions, Application – A Logic based financial advisor.

UNIT - 2

Introduction to structure and Strategies for State Space search, Graph theory, Strategies for state space search, Using the State Space to Represent Reasoning with the Predicate calculus (Sate space description of a logical system, AND/OR Graph). Heuristic Search : introduction, Hill-Climbing and Dynamic Programming, The Best-first Search Algorithm, Admissibility, Monotonicity and informedness, Using Heuristics in Games.

UNIT - 3

Building Control Algorithm for Statespace search – Introduction, Production Systems, The blackboard architecture for Problem solving. Knowledge Representation – Issues, History of AI representational schemes, Conceptual Graphs, Alternatives to explicit Representation, Agent based and distributed problem solving.

UNIT - 4

Strong Method Problem Solving – Introduction, Overview of Expert System Technology, Rule Based Expert system, Model -Based, Case-Based and Hybrid Systems (Introduction to Model based reasoning, Introduction to Case Based Reasoning, Hybrid design), Introduction to Planning. Reasoning in Uncertain Situation – introduction, logic based Abductive Inference. Introduction to PROLOG , Syntax for predicate Calculus programming, ADTs, A production system example.

UNIT - 5

Machine Learning: Symbol Based – Introduction, Frame –work. The ID3 Decision tree Induction algorithm. Inductive bias and Learnability, Knowledge and Learning, Unsupervised learning, Reinforcement Learning, Machine Learning : Connectionist – Introduction, foundations, Perceptron learning. Machine learning : Social and emergent: Models, The Genetic Algorithm, Artificial Life and Social based Learning.

Text book :

1. George F Luger, Artificial Intelligence – Structures and Strategies for Complex problem solving, 5thEdn, pearson.

Reference Books:

1. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill.
2. S. Russel and p. Norvig, Artificial intelligence – A Modern Approach, 3rdEdn, pearson
3. D W Patterson, introduction to Artificial Intelligence and Expert Systems, PHI, 1990

MCS4C18: SOFT COMPUTING TECHNIQUES

UNIT-1

Introduction – Soft Computing, Requirements, Areas, Applications. Neural Networks: Single layer Perceptrons, Back Propagation networks, Architecture of Back propagation(BP) Networks, Back propagation Learning, Associative Memory, Adaptive Resonance theory and Self Organizing Map, Applications.

UNIT-2

Fuzzy Systems - Fuzzy Set theory, Fuzzy versus Crisp set, Fuzzy Relation, Fuzzification, Minmax Composition, Defuzzification Method, Fuzzy Logic, Fuzzy Rule based systems, Predicate logic, Fuzzy Decision Making, Fuzzy Control Systems, Fuzzy Classification.

UNIT-3

Genetic Algorithm - Working Principle, Various Encoding methods, Fitness function, GA Operators- Reproduction, Crossover, Mutation, Convergence of GA, Bit wise operation in GA, Multi-level Optimization. GA based Back propagation Networks: GA based Weight Determination, K - factor determination in Columns.

UNIT-4

Hybrid Systems: Sequential Hybrid Systems, Auxiliary Hybrid Systems, Embedded Hybrid Systems, Neuro-Fuzzy Hybrid Systems, Neuro-Genetic Hybrid Systems, Fuzzy-Genetic Hybrid Systems.

UNIT-5

Evolutionary Computation - Survival of the Fittest - Fitness Computation – Crossover – Mutation – Reproduction - Rank space Method. Case Studies: Applications of soft computing.

Text books:

1. N.P.Padhy, S.P.Simon, “Soft Computing with MATLAB Programming”, Oxford University Press, 2015.
2. S.N.Sivanandam , S.N.Deepa, “Principles of Soft Computing”, Wiley India Pvt. Ltd., 2nd Edition, 2011.
3. S.Rajasekaran, G.A.Vijayalakshmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithm, Synthesis and Applications “, PHI Learning Pvt. Ltd., 2017.
4. Genetic Algorithms: Search and Optimization, E. Goldberg.
5. Neuro-Fuzzy Systems, Chin Teng Lin, C. S. George Lee, PHI

MCS4C19: MACHINE LEARNING

UNIT-1

Basic Principles: Introduction, Experimental Evaluation: Over-fitting, Cross-Validation. PAC learning. Sample complexity, VC-dimension, linear classification; Classification errors; Regression Techniques.

UNIT-2

Supervised Learning: Gradient Descent, SVM and Multiclass SVM, Learning with structured output spaces: struct SVM. Algorithms: Cutting plane algorithm, Graph based semi supervised learning, Back-propagation, Layers, State, Memory, Representations.

UNIT-3

Unsupervised Learning: Clustering, Dimension reduction, Expectation Maximization, Mixture of Gaussians, Hidden Markov Models, Anomaly detection.

UNIT-4

Probabilistic Models: Probabilistic Models: Maximum Likelihood Estimation, MAP, Probabilistic Principal Component Analysis, Latent Dirichlet allocation.

UNIT-5

Kernel Methods and Latent Space Models: Basics, Gaussian Processes, Kernels on Strings, trees, graphs, Latent Space Models: Independent Component Analysis.

Lab Experiments

- Experiments based on validation of models, regression and classification.
- Experiments on Gradient descent and stochastic gradient descent.
- Case studies using SVN and Multiclass SVM
- Experiments on Back propagation
- Implementations of Dimension reduction, EM algorithm and HMM
- Implementation of MAP, PCA, LDA, Kernel methods.
- Apply of machine learning methods to real word applications (Case studies)

Text Books :

1. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
2. Machine Learning: A Probabilistic Perspective, Kevin P Murphy, MIT Press.
3. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer 2006.
4. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern Classification. John Wiley & Sons, 2006.
5. Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer 2009.
6. MacKay, David. Information Theory, Inference, and Learning Algorithms. Cambridge, UK: Cambridge University Press, 2003.

MCS4C20: DEEP LEARNING

UNIT-1

Basics-Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Threshold logic, Linear Perceptron, Multilayer Perceptron, Perceptron Learning Algorithm, Linear separability. Convergence theorem for Perceptron Learning Algorithm. Deep Neural Networks: Difficulty of training deep neural networks, Greedy layer wise training. Better Training of Neural Networks: Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural network

UNIT-2

Deep Feed forward Networks- Gradient Descent, hidden units, Back propagation, Regularization for deep learning, Optimization for training deep models-Empirical Risk Minimization.

UNIT-3

Convolutional Neural Networks:- LeNet, AlexNet. Recurrent Neural Networks: Back propagation through time, Bidirectional RNNs, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs

UNIT-4

Auto encoder- Generative models: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

UNIT-5

Recent trends: Variational Auto encoders, Generative Adversarial Networks, Multi-task Deep Learning, Multi-view Deep Learning Applications: Computer Vision, NLP, Speech.

Text Books

1. Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016.
2. Yuxi (Hayden), Liu and Savansh Mehta, “Hands -on Deep Learning Architectures with Python”, Packt, 2019.
3. Sudharsan Ravichandiran,“Hands -on Deep Learning Algorithms with Python”,Packt, 2019.
4. Sandro Skansi, “Introduction to Deep Learning from Logical calculus to Artificial Intelligence”, Springer , 2018.
5. Nikhil Ketkar, Deep Learning with Python- A Hands on Introduction,Apress, 2017.
6. Neural Networks: A Systematic Introduction, Raúl Rojas, 1996
7. Pattern Recognition and Machine Learning, Christopher Bishop, 2007
8. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville 9.
9. Neural Networks and Deep Learning by Michael Nielson.

Syllabus – Lab Courses

MCS1P01 Lab I (IP/OS)

Hours/Week : 8 +2 **Credit : 3**

Section A – Introduction to Programming (C & C++) (5 +2 Hours / Week)

Faculty-in-charge shall prepare a list of experiments at the beginning of the semester. For the ESE, question will be selected from this list. All exercises must be done under Linux environment.

Sr No	Topic /Description	Minimum Number of Programs
1	If and switch statements.	1
2	Using Loops	2
3	String manipulation	1
4	Search	1
5	Sort	1
6	Matrix operations	2
7	Functions	2
8	Pointer	2
9	Structure / Union	2
10	Class, constructors, destructors – simple programs	5
11	Friend function	1
12	Friend class	1
13	Function overloading	1
14	Operator overloading	1
15	Programs illustrating Inheritance , Virtual base class, Polymorphism, virtual functions	3
16	C++ Files	1
17	STL	1
18	C++ I/O based exercises	2
	Total	30

Section B : Operating System (3 hours per week)

Faculty-in-charge shall prepare a list of experiments at the beginning of the semester. Use C/C++ for high level programming.

i. Linux basic and essential commands
ii. Editors in Linux
iii. Linux Shell programming : minimum 5 programs
iv. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
v. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
vi. Implement memory management schemes (Minimum two schemes).

ESE Scheme of evaluation

1.	Record of work done duly certified	:	10
2.	C program	:	20
3.	C++ program	:	20
4.	Viva based on C/C++	:	10
5.	Linux shell program	:	10
6.	Viva based on Linux lab assignments	:	10
	Total	**:**	**80**

For 2,3 and 5:
Program writing : 5
Execution without errors: 5
Output : 5
Questions based on the program and/or Modification: 5

MCS2P02 Lab II (Java/DS/DBMS)

Hours/Week :7 +1 **Credit : 3**

Section A– JavaProgramming (2 +1 Hours per week)

Faculty-in-charge shall prepare a list of experiments, based on the topics specified below, at the beginning of the semester. For the ESE, question will be selected from this list. All exercises must be done under Linux environment.

Sr No	Topic / Description	Minimum Number of Programs
1	Simple programs employing class – covering basic class concepts.	2
2	Nested and Inner class	1
3	String manipulation	1
4	Command line arguments	1
5	Inheritance, Method overloading /overriding, Abstract class	3
6	Packages – Covering important concepts in package	2
7	Exception handling	1
8	Thread and multi-threaded applications	3
9	Applet	2
10	AWT	2
11	Event driven programs	2
12	Database connectivity - jdbc	2
	Total	22

Section B : Data Structures (3 hours per week)

Faculty-in-charge shall prepare a list of experiments at the beginning of the semester. Use C++ under Linux for implementation. Employ class concepts and features of C++ in all exercises.

Sr No	Topic / Description	Minimum Number of Programs
1	Polynomial representation and operations	1
2	Sparse matrix representation and operations	1

3	Singly linked list	3
4	Doubly linked list	1
5	Stack/Queue with SLL	1
6	Infix to postfix	1
7	Postfix evaluation	1
8	Circular array queue	1
9	Create Binary tree	1
10	Recursive tree traversal	1
11	Non-recursive tree traversal	1
12	Threaded Binary tree	1
13	Binary search tree	1
14	AVL tree / Hashing	1
15	Dijkstra's algorithm/ Prim's algorithm/Kruskal	1
16	Quick sort / merge sort	2
17	Heap sort / Warshal's algorithm	1
	Total	20

Section C : Database Management Systems (2 Hours per week)

Faculty-in-charge shall prepare a list of experiments at the beginning of the semester. Use PostgrSQL for the lab exercises. ESE questions will be set based on the list provided by the faculty-in-charge. Exercises shall include the following components:

1. Create databases and tables, different types of Constraints, SQL queries to add/delete/retrieve data.
2. SQL queries : Update, modify, Alter, Join, nested queries etc.
3. Index, operators and functions, views, arrays, transactions, cursors, triggers, etc.
4. PostgreSQL Administration
5. PostgreSQL Programming - Pl/pgSQL
6. Case study – design of database for a simple application like payroll and its implementation.

ESE Scheme of evaluation

1. Record of work done duly certified	:	10
2. Java or DS program	:	30
3. DBMS	:	20

4.	Viva	: 20
	Total	**: 80**

For 2:

Program writing :7

- Execution without errors: 8
- Output/Correctness : 8
- Questions based on the program and/or Modification: 7

For 3

Program/Query writing : 5

- Execution without errors: 5
- Output/Correctness : 5
- Questions based on the program/problem and/or Modification: 5

MCS3P04 Lab III (CG/NP&A/SP&CD)

Hours/Week :6 +2 **Credit : 3**

Section A – Computer Graphics(2 +1 Hours per week)

Faculty-in-charge shall prepare a list of experiments, based on the topics specified below, at the beginning of the semester. For the ESE, question will be selected from this list. All exercises must be done using OpenGL (under Windows or linux).

Sr No	Topic / Description	Minimum Number of Programs
1	OpenGL Point and Line functions with different attributes Simple OpenGL programs with I/O and Mouse support	2
2	Line drawing algorithms	2
3	Circle Drawing	1
4	Line Clipping	1
5	Polygon Clipping	1
6	2D transformations	1
7	3D View based	2
8	3D transformations	1
9	3D object representations	1
10	Visible surface detection methods	1
11	Illumination / Rendering	1
	Total	14

Section B : Network Programming and System Administration (2+1 hours per week)

Faculty-in-charge shall prepare adetailed descriptionof experiments.

Sr No	Topic / Description
1	Configuration : FTP, TFTP, IP address
2	Configuring NIS
3	Configuring DHCP.
4	Configuring SAMBA server.
5	Setting Domain Name Services.
6	SMTP and POP3
7	TCP chat program.
8	UDP chat program.
9	Socket program.
10	Configuring NFS.
11	LILO configuration
12	Crontab, at, Batch.
13	Kernel modules
14	Run levels
15	TCP - wrappers
16	Changing file permission, group and owner.
17	Syslog.conf
18	Backup (tar, cpio, dd etc.)
19	Rescue operations.

Section C : System Programming and Compiler Design (2 Hours per week)
Implement the following:

Sr No	Topic / Description
1	Simple Assembler
2	Using Lex and Yacc - validation of expressions, validation of variable names, implementation of calculator.
3	Implementing any three parsing algorithm
4	Implement Symbol Table
5	Intermediate code Generator
6	Code Optimizer.

ESE Scheme of evaluation

1. Record of work done duly certified	:	10
2. CG program	:	20
3. NP & A	:	20
4. Viva based on CG and NP&A exercises	:	15
5. Execution of selected exercise from SP&CD and Viva based on that	:	15

Total : 80

For 2 and 3
Program writing : 5
Execution without errors: 5
Output/Correctness : 5
Questions based on the program/problem and/or Modification: 5

MCS2P03 Case Study I and MCS3P05 Case Study II

Hours/Week : 3+2 **Credit : 2**

The objective of the course is to inculcate self-learning skill in mastering software development tools. The department shall select one or more of Development tools such as .Net, Python, HTML/PHP/JavaScript, Android and Mathlab. The teacher-in-charge shall give an overview of the tool and if required arrange for lectures by external experts. The teacher may also help the students to find online tutorials/courses. A set of lab assignment shall be prepared by the teacher. Each student is expected to solve a problem using the tool(s) selected. Unlike project work, the focus should be on coding and testing of programs. A report with the statement of problem, description of solution, code and output is to be submitted for the external evaluation.

ESE Scheme of evaluation

1. Report duly certified : 10
2. Demonstration of the software : 15
3. Viva based on the Tool and Software : 15

MODEL QUESTIONS

I &II SEMESTER

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 1			**Course : MCS1C01 DISCRETE MATHEMATICS**				
Section A				**Section B**			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION		
UNIT	**SECTION A**	**SECTION B**
1	3	1
2	3	1
3	2	1
4	2	1
5	2	1

Note:

At least 60% of the questions should be problems.

The questions shall include simple/direct questions (approximately 30%), Average / moderate difficulty (40%) and Challenging / difficult questions 930%)

Model Question

FIRST SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS1C01 DISCRETE MATHEMATICS

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1.Write the following statements in symbolic form:

i. x is the father of mother of y.

ii. All men are mortal

2.Construct the truth table for $(P \rightarrow Q) \wedge (Q \rightarrow P)$

3. Show that $P \Rightarrow (P \vee Q)$

4. Show that $A \bigcup_{i=1}^{n} Bi = \bigcap_{i=1}^{n} (A - Bi)$

5. Let X={1,2,3,4} and R be the relation defined on the set X as R={<x,y>,x≤y}.Write the relation matrix.

6. Find the power set of {1,2,3,{1,2,3})

7. Find the value of n, if P(n,7)=12P(n,5).

8.Neethu has 5 friends, in how many ways can she invite two or more of them to a tea party.

9. Four dice are thrown simultaneously. Find the probability that all of them show the same face.

10. What is a subring? Find subring of **<I, +, . >,** where I is the set of Integers.

11. Define Bipartite graph. Give example.

12. Define the terms walk, path, trail and circuit.

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. a) Explain the different measures used in accessing the performance of computer systems.

OR

b) i. Explain the role of stacks.

ii. Explain instruction sequencing. (4+6)

14. a) Give a detailed account of Interrupts in relation to IO operations.

OR

b) Give a detailed account of USB standard.

15. a) Explain 3-bus organization of processors.

OR

b) With suitable example explain Booth algorithm.

16. a) Explain the organization and working of virtual memory.

OR

b) Discuss different cache mapping techniques.

17. a) Discuss the major issues related to pipelining.

OR

b) Explain the basic organization and advantages of Vector processing and shared memory multiprocessors.

(5 x 10 = 50 marks)

<table>
<tr><th colspan="8">INSTRUCTIONS TO QUESTION PAPER SETTER</th></tr>
<tr><td colspan="8"><b>Semester : 1</b> <b>Course : MCS1C02COMPUTER ORGANIZATION AND ARCHITECTURE</b></td></tr>
<tr><td colspan="4"><b>Section A</b></td><td colspan="2"><b>Section B</b></td><td colspan="2"></td></tr>
<tr><td><b>Total no of questions</b></td><td><b>Mark per question</b></td><td><b>No of questions to be answered</b></td><td><b>Time per question</b></td><td><b>Total no of questions</b></td><td><b>Mark per question</b></td><td><b>No of questions to be answered</b></td><td><b>Time per question</b></td></tr>
<tr><td>12</td><td>3</td><td>10
10 x 3 = 30</td><td>5 to 8 minutes</td><td>5 either-or questions - each containing two parts - (a) and (b)</td><td>10 marks per each part
5 x 10 = 50</td><td>All five questions selecting one part from each question</td><td>20 to 25 minutes</td></tr>
<tr><td colspan="8"><b>UNIT WISE DISTRIBUTION</b></td></tr>
<tr><td colspan="2"><b>UNIT</b></td><td colspan="3"><b>SECTION A</b></td><td colspan="3"><b>SECTION B</b></td></tr>
<tr><td colspan="2">1</td><td colspan="3">2</td><td colspan="3">1</td></tr>
<tr><td colspan="2">2</td><td colspan="3">2</td><td colspan="3">1</td></tr>
<tr><td colspan="2">3</td><td colspan="3">3</td><td colspan="3">1</td></tr>
<tr><td colspan="2">4</td><td colspan="3">3</td><td colspan="3">1</td></tr>
<tr><td colspan="2">5</td><td colspan="3">2</td><td colspan="3">1</td></tr>
</table>

Model Question

FIRST SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS1C02 COMPUTER ORGANIZATION AND ARCHITECTURE

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. With suitable example explain 2’s complement scheme for signed integer representation.
2. Compare CISC and RISC.
3. What do you mean by bus arbitration?
4. Differentiate between program controlled IO and Interrupt driven IO.
5. Explain fetch-execute cycle.
6. How will store floating point numbers?
7. Give the basic principle of array multiplier.
8. Give the memory hierarchy.
9. What are the different types of DRAMs?
10. Explain any one page replacement strategy used in Cache memory.
11. Explain the idea of hardware multithreading.
12. List the advantages of pipeline processing.

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. a)Explain the different measures used in accessing the performance of computer systems.

OR

b) i. Explain the role of stacks.

ii. Explain instruction sequencing. (4+6)

14. a) Give a detailed account of Interrupts in relation to IO operations.

OR

b) Give a detailed account of USB standard.

15. a) Explain 3-bus organization of processors.

OR

b) With suitable example explain Booth algorithm.

16. a) Explain the organization and working of virtual memory.

OR

b) Discuss different cache mapping techniques.

17. a) Discuss the major issues related to pipelining.

OR

b) Explain the basic organization and advantages of Vector processing and shared memory multiprocessors.

(5 x 10 = 50 marks)

\

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 1			Course : MCS1C03 DIGITAL SYSTEMS & MICROPROCESSORS				
Section A				Section B			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION

UNIT	SECTION A	SECTION B
1	3	1
2	3	1
3	2	1
4	2	1
5	2	1

Model Question

FIRST SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS1C03 DIGITAL SYSTEMS & MICROPROCESSORS

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. Show that complement of EX-OR is equal to its dual.
2. Discuss the main differences between SOP and POS circuits and their implementation.
3. What is the underlying principle of K-Map and Tabular method simplification? Illustrate with examples.
4. Show how a 2 to 4 decoder circuit can be modified to 1 to 4 Dmux.
5. What is Master-Slave Flip flop?
6. What are the disadvantages of ripple counters and how they are overcome?
7. What is addressing mode in a processor. List the different addressing modes in 8085 processor?
8. Explain the complete functioning of the following instructions in 8085 processor:
   i) ADD B ii) RST1
9. Give the Flag register of 8686.
10. What are assembler directives?
11. Explain the protection mechanism used in advanced processors.
12. What is the concept of page mode in 80386 processor?

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. a) i. Simplify the following function using K-Map and draw the simplified circuit.
$F(w,x,y,z)=\Sigma(0,1,2,3,7,8,10)$ and $d(w,x,y,z)=\Sigma(5,6,11,15)$
ii. Draw the circuit of a 4 bit parallel adder. 6+4

OR

b)i. Explain how parity is used for error checking. Show a scheme to generate even parity and transmission and checking at receiving end.

ii. Simplify the following boolean function using Tabular method finding the essential prime implicants and draw the circuit:
$F(A,B,C,D) = \Sigma(0,2,3,5,7,8,10,11,14,15)$ 4+6

14. a) Discuss the design of a Mod 10 counter.

OR

b) i. With the help of a block diagram, explain the working of serial to parallel shift register.

ii. With the help of a block diagram, explain the working of JK flip flop. 6+4

15. a) Explain the architecture of 8085 microprocessor.

OR

b) i. Explain the different operations taking place in each machine cycles while a CALL instruction is executed

ii. Draw the structure of register corresponding to SIM instruction and explain each bit.

6+4

16. a) Explain the architecture of 8086 microprocessor.

OR

b)Explain organization and working of 8255.

17. a) i. Compare 386 and 486 processors.
ii. Explain the memory management unit of advanced processors. 4+6

OR

b) Discuss the special features of Pentium processors.

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 1 Course : MCS1C04 OPERATING SYSTEMS							
Section A				Section B			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION

UNIT	SECTION A	SECTION B
1	2	1
2	3	1
3	3	1
4	2	1
5	2	1

Model Question

FIRST SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS1C04 OPERATING SYSTEMS

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. List the salient features of Real-time systems.
2. What do you mean by system calls? Give any two examples.
3. Explain the concept of multithreading.
4. Define “Critical section” and “Semaphores”.
5. Explain the CPU scheduling criteria.
6. Explain Thrashing.
7. Explain the relevance of Virtual memory concept in modern operating systems.
8. Compare Windows and Linux directory structure.
9. Explain the basic principle of RAID.
10. Explain “STREAMS”.
11. List the distinguishing features of Distributed Operating systems.
12. Explain the terms Stateful and stateless services.

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. **a)** Give a detailed account of Operating system services.

OR

b) Discuss the structure of a typical operating system.

14. a) Explain any two preemptive and any one non-preemptive scheduling algorithm. Illustrate the algorithms taking suitable example.

OR

b) What are the methods for handling deadlock? How will you prevent deadlock?

15. a) Explain the need for paging and segmentation. Discuss the different page replacement algorithms.

OR

b) Discuss : i. Free space management ii. NFS

16. a) Discuss Kernel I/O subsystems.

OR

b) Give a detailed account of Disk scheduling and Disk structure.

17. a) Explain the design issues of distributed systems.

OR

b) Give a comprehensive account of Protection.

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 1 Course : MCS1C05 INTRODUCTION TO PROGRAMMING							
Section A				Section B			
Total no of questions	Mark per question	No of questions to be answered	Time per question	Total no of questions	Mark per question	No of questions to be answered	Time per question
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION

UNIT	SECTION A	SECTION B
1	3	1
2	3	1
3	2	1
4	2	1
5	2	1

Model Question

FIRST SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS1C05 INTRODUCTION TO PROGRAMMING

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. Give the syntax of for loop and while Loop.
2. List and explain any three string functions.
3. Discuss hierarchy of operations in c expressions.
4. Write a function to find the length of a string.
5. Differentiate structure and Union.
6. What do you mean by preprocessor directives? Give examples.
7. What is a constructor? What are the different types of constructors in C++?
8. What do you mean by dynamic allocation of memory?
9. Give an example of operator overloading.
10. What is a virtual function?
11. What is a stream?
12. Explain the term “Name spaces”.

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. a) Write a complete c program to multiply two matrices.

OR

b) i. Write a c program to search a name in an array of n names.

ii. With suitable examples, explain the different decision making and branching constructs in c. 6+4

14. a) What is a pointer? What are the operations performed on pointers? Explain with example how pointers can be used to access array elements.

OR

b) i. Declare a structure for storing student information (make your own assumptions). Write a function to read details of a student into a structure variable.

ii. Explain the different parameter passing mechanisms with suitable examples.

15. a) Explain friend functions and friend class with suitable examples.

OR

b) Declare a class for storing the details of books. Include suitable constructor and methods for reading and printing the details.

16. a) With suitable example explain function overloading.

OR

b)With suitable examples explain different types of inheritance supported in C++.

17. a) Give detailed account of file processing n C++.

OR

b) Give a detailed account of STL.

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 2				Course : MCS2C06 JAVA PROGRAMMING			
Section A				Section B			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION		
UNIT	**SECTION A**	**SECTION B**
1	2	1
2	3	1
3	3	1
4	2	1
5	2	1

Model Question

SECOND SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS2C06 JAVA PROGRAMMING

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. Explain JVM.
2. Explain type casting in Java.
3. With suitable example explain constructors.
4. What do you mean by method overloading?
5. What is the significance of Garbage collection?
6. What is a package?
7. What do you mean by runnable interface?
8. What is a deadlock in thread management?
9. What is an applet?
10. What is AWT?
11. What is an event?
12. What is a stored procedure?

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. (a) Give a detailed account of control statements in Java.

Or

(b)i. List and explain important features of Java.

ii. Write a Java program to search a name in an array of names. (5+5)

14. (a) i. Differentiate between Class and Interface.

ii. Describe the following methods: replace, compareTo and charAt. (5+5)

Or

(b) With suitable examples explain Inheritance.

15. (a) Give a detailed account of exception handling in Java.

Or

(b) i. Explain how to create a package with suitable example.

ii. Write a program to create two threads, one thread will print odd numbers and second thread will print even numbers between 1 to 20 numbers. (5+5)

16. (a) Explain Applet life cycle.

Or

(b) Discuss AWT controls.

17. (a) Give a detailed account of event handling.

Or

(b) Discuss JDBC architecture.

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 2 Course : MCS2C07 Data Structures & algorithms							
Section A				Section B			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION		
UNIT	**SECTION A**	**SECTION B**
1	3	1
2	3	1
3	2	1
4	2	1
5	2	1

Model Question

SECOND SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS2C07 DATA STRUCTURES & ALGORITHMS

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. Write a function to concatenate tow singly Linked List.
2. Explain singly Linked List representation of polynomials.
3. With suitable example, explain the significance of Big Oh in algorithm analysis.
4. Give a class declaration for array based stack.
5. Define Tree and Binary Tree.
6. What is a priority Queue? Give any one application of it.
7. Differentiate red-Black tree and AVL tree.
8. Write a recursive function for the inorder traversal of a binary tree.
9. Differentiate Depth fist and Depth first approaches in graph traversal.
10. Discuss any two graph representation schemes.
11. Explain divide-and-Conquer strategy.
12. Give the basic principle of Quick sort.

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. **a)** Explain a scheme of representing Sparse Matrices. Write and explain a function to add two sparse matrices represented with your scheme. State your assumptions.

OR

b) i. Write a recursive function to reverse a Singly Linked List.

ii. Write a function to delete the ith node (if exist) froma singly Linked List.

iii. Write a function to delete first node from a Doubly Linked List. (3+4+3)

14. a) Explain Infix, Postfix and Prefix notations with examples. Write a complete program to convert an infix expression to postfix. Include class declarations and functions for stack operations.

OR

b) Write a Non-recursive function to traverse a Binary tree inorder. Include class declarations, functions for stack/Queue if employed. Comment on it time complexity.

15. a) What is a threaded Binary tree? Give its advantage. Write a function for inorder traversal of an Inorder Threaded Binary tree. Include appropriate class declarations.

OR

b) Explain Hashing. Discuss any three Collision Resolution techniques. Highlight advantages/limitations of each technique.

16. a) Explain shortest path problem. Write and illustrate with suitable example Dijkstra's algorithm.

OR

b) Define graph? List and explain any two applications of graph. Write a and explain Prim's algorithm. Illustrate with suitable example.

17. a) Write and explain necessary functions for Heap sort.

OR

b) Explain the principle of dynamic program. Write and explain Warshal's algorithm.

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 2			**Course : MCS2C08 DATABASE MANAGEMENT SYSTEMS**				
Section A				**Section B**			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION		
UNIT	**SECTION A**	**SECTION B**
1	3	1
2	3	1
3	2	1
4	2	1
5	2	1

Model Question

SECOND SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS2C08 DATABASE MANAGEMENT SYSTEMS

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. Given the following relations :

   EMP (Name, Eno, Deptno, Salary)
   DEPT (Deptno, Dname,Location)

   Write a query in SQL to find the name of the employee of each department who is getting highest salary.
2. Explain join operation in relational algebra.
3. Explain the difference between procedural and non-procedural DML
4. Differentiate between primary key, candidate key and super Key in ER model.
5. What is UML?
6. Explain the terms Assertion and Triggers.
7. Differentiate between Indexing and Hashing.
8. What do you mean by a transaction?
9. What do you mean by Distributed databases?
10. List any three differences between RDBMS and OODBMS.
11. List the data types in PostgreSQL.
12. What do you meant by aggregate function in pgSql

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. a) i. Discuss the salient features of Relational Model.

ii. Consider the following relation schemes:

Project (Project#, Project_name, chief_architect)
Employee (Emp#, Empname)
Assigned_To (Project#, Emp#)

Give expression in Tuple Relational calculus and Domain Relational calculus for each of the queries below:

(i) Get the employee numbers of employees who work on all projects.
(ii) Get the employee numbers of employees who do not work on the COMP123 project. (5+5)

OR

b) Explain the basic structure of SQL. With suitable example explain Set operations in SQL.

14. a)Give a detailed account of different Normal forms.

OR

b) Construct an ER diagram with all major components for a banking enterprise with entity sets customer, branch, loan, payment, account etc along with your own assumptions.

15. a) Explain the essential properties of transactions. Explain Concurrency control with examples.

OR

b) What is the significance of Hashing in DBMs? Give different Hashing schemes employed in DBMS.

16. a) Give a detailed account of Data warehousing.

OR

b)Discuss basic concepts, organization, advantages and major issues related to parallel databases

17. a) Give a detailed account of PostgreSQL administration.

OR

b) with suitable example explain postgreSQL programming.

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 2				Course : MCS2C09 COMPUTER NETWORKS			
Section A				Section B			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION

UNIT	SECTION A	SECTION B
1	3	1
2	2	1
3	2	1
4	2	1
5	3	1

Model Question

SECOND SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS2C09 COMPUTER NETWORKS

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. List four main applications of Internet.
2. What are the features of an optical fiber?
3. List the services provided by Data Link Layer.
4. Write a note on simplex stop-and-wait protocol.
5. What is fast Ethernet?
6. Compare and contrast Pure ALOHA and Slotted ALOHA.
7. Write short notes on hierarchical routing.
8. Explain Flooding.
9. What is Cryptography?
10. Write short notes on Simple Transport Protocol.
11. Write the functions of LLC.
12. What is the difference between connection oriented communication and connectionless communication.

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. (a) Discuss OSI reference model as network architecture.

Or

(b)Discuss LAN, WAN, MAN with respect to speed, coverage (area) and topology.

14. (a) Explain the sliding window protocol in detail.

Or

(b) Compare and contrast error detection with error correction.

15. (a) Discuss the frame format for 802.3 LAN.

Or

(b) Discuss CSMA/CD protocol used in LAN .

16. (a) What is an IP address? Explain the different classifications of IP address.

Or

(b) Discuss any four methods of congestion control in datagram subnets.

17. (a) Explain about the TCP header and working of the TCP protocol.

Or

(b) What is UDP? Explain the structure of UDP header?

(5 x 10 = 50 marks)

INSTRUCTIONS TO QUESTION PAPER SETTER							
Semester : 2		**Course : MCS2C10 FORMAL LANGUAGES AND FINITE AUTOMATA**					
Section A				**Section B**			
Total no of questions	**Mark per question**	**No of questions to be answered**	**Time per question**	**Total no of questions**	**Mark per question**	**No of questions to be answered**	**Time per question**
12	3	10 10 x 3 = 30	5 to 8 minutes	5 either-or questions - each containing two parts - (a) and (b)	10 marks per each part 5 x 10 = 50	All five questions selecting one part from each question	20 to 25 minutes

UNIT WISE DISTRIBUTION		
UNIT	**SECTION A**	**SECTION B**
1	3	1
2	3	1
3	3	1
4	2	1
5	1	1

Model Question

SECOND SEMESTER M Sc Computer Science DEGREE EXAMINATION

MCS2C10 FORMAL LANGUAGES AND FINITE AUTOMATA

Time: 3 Hrs **Max Marks: 80**

Section A

Answer any ten questions. Each question carries three marks.

1. Define NFA and DFA.
2. Differentiate between transducers and acceptors.
3. Define Language and Grammar.
4. Define Regular grammar. Give example.
5. Write Regular expression for the language on {0,1} '*all strings ending in 01*'.
6. Explain Parsing with suitable example.
7. Define CNF and GNF.
8. Differentiate between PDA and DFA.
9. Define Deterministic PDA.
10. What do you mean by a context free Language? Give example.
11. Define standard Turing machine.
12. What is a Linear Bounded Automata?

(10 x 3 = 30 marks)

Section B

Answer all questions. Each question carries ten marks.

13. (a) i. Prove that. If a regular language L is accepted by an NFA then there exists a DFA to accept L.

ii. Find a DFA for the language on $\Sigma = \{a, b\}$ $L = \{w: |w| \bmod 2 = 0\}$ (7+3)

Or

(b)i. With suitable example illustrate how number of states in a Finite Automaton can be reduced.

ii. Find a grammar for $\Sigma = \{a, b\}$ that generate the sets of all string's with no more than two a's. (7+3)

14. (a) i. Prove that $L=\{0^i1^j / i>j\}$ is not regular using pumping lemma.

ii. Define derivation tree. With an example explain leftmost and rightmost derivation.

(6+4)

Or

(b) i. Define regular language and regular grammar.

ii. Define context free grammar. With suitable example explain ambiguity in grammar.

(4+6)

15. (a) i. Remove useless and unit productions from the grammar S→Aa/ B, B→A/bb, A→a/bc/B.ii. Convert the grammar with productions S→ABa, A→aab, B→Ac. to Chomsky Normal Form.

(5+5)

Or

(b) i. Construct an NPDA for the language. $L=\{w Є \{a,b\}^* : n_a(w) = n_b(w)\}$.

ii. Prove that for any context free language L, there exists an NPDA M such that L=L(M).

(5+5)

16. (a) State and prove pumping lemma for Context free Languages.

Or

(b)With suitable example explain how Turing machine can be implemented as a Transducer.

17. (a) Explain i. Nondeterministic Turing Machine ii. Universal Turing Machine.

Or

(b) Discuss limits of algorithmic computation.

(5 x 10 = 50 marks)