GMRIT

Academic Regulations, Curriculum and Syllabi 2021

B. Tech. CSE Specialization
Artificial Intelligence & Data Science

(Duration of Study : 4 years)

GMRIT
Training Tomorrow's Engineers Today

Department of Computer Science and Engineering
GMR Institute of Technology
Rajam, Andhra Pradesh
(An Autonomous Institute Affiliated to JNTU Kakinada, AP)
NBA Accredited and NAAC Accredited

Academic Rules and Regulations

2021

Undergraduate Programs

GMRIT
Training Tomorrow's
Engineers Today

GMR Institute of Technology
Rajam 532 127, AndhraPradesh
(An Autonomous Institute, Affiliated to JNTUK, Kakinada, AP)
Accredited by NAAC & NBA

The Vision of GMRIT

- To be among the most preferred institutions for engineering and technological education in thecountry
- An institution that will bring out the best from its students, faculty and staff – to learn, to achieve, to compete and to grow – among the verybest
- An institution where ethics, excellence and excitement will be the work religion, while research, innovation and impact, the workculture

The Mission of GMRIT

- To turnout disciplined and competent engineers with sound work and lifeethics
- To implement outcome based education in an IT-enabledenvironment
- To encourage all-round rigor and instill a spirit of enquiry and critical thinking among students, faculty andstaff
- To develop teaching, research and consulting environment in collaboration with industry and otherinstitutions

Department Vision

To be a nationally most preferred department of learning for students and teachers alike,with dual commitment to research and serving students in an atmosphere of innovationand critical thinking.

Department Mission

1. To provide high-quality education in Computer Science Engineering to prepare the graduates for a rewarding career in Computer Science Engineering and related industries, in tune with evolving needs of theindustry.
2. To prepare the students to become thinking professionals and good citizens who would apply their knowledge critically and innovatively to solve professional and social problems.

Program Educational Objectives

PEO1: To produce the competent software engineers as team players in industry andallied fields providing viable solutions.

PEO2: Adopt contemporary technologies for dynamic industry requirements with self-paced learning providing scope for advanced research.

PEO3: Nurture professionalism with soft skills, managerial & leadership skills and Ethical values

Program Outcomes

Engineering graduate will be able to

PO 1: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. (Engineering knowledge)

PO 2: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. (Problem analysis)

PO 3: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. (Design/development ofsolutions)

PO 4: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. (Conduct investigations of complex problems)

PO 5: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. (Modern tool usage)

PO 6: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. (The engineer and society)

PO 7: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. (Environment and sustainability)

PO 8: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. (Ethics)

PO 9: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. (Individual and team work)

PO 10: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. (Communication)

PO 11: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. (Project management and finance)

PO 12: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. (Life-long learning)

PSO1: Ability to apply the software engineering principles to meet automation of theprocess and service industries apart from the community utilities.
(Program Specific)

PSO2: Ability to design, develop and implement management systems, E-Commercetools and WebApps for product development.(Program Specific)

Department of CSE-AI&DS

Minimum Credits to be earned: 160 (for Regular Students)
123 (for Lateral Entry Students)

S.No	Course Code	Course Name	POs	L	T	P	C
First Semester							
1	21HSX01	Communicative English	10, 12	2	-	-	2
2	21MAX01	Engineering Mathematics I	1	3	-	-	3
3	21PYX01 21CYX01	Engineering Physics / Engineering Chemistry	1 / 1	3/3	-	-	3/3
4	21BEX01 21BEX06	Basics of Engineering / IT Workshop	1,12/1,12	3/ -	-	-/ 3	3/1.5
5	21BEX02	Problem Solving and Programming Skills	1, 12	3	-	-	3
6	21BEX03	Problem Solving and Programming Skills Lab	4	-	-	3	1.5
7	21BEX04/ 21BEX05	Engineering Drawing / Engineering Workshop	1,5,10/1,9,10	-	-	3/3	1.5/1.5
8	21PYX02/ 21CYX02	Engineering Physics Lab /Engineering Chemistry Lab	4/4	-	-	3/3	1.5
9	21HSX02/-	Communicative English Lab/-	10,12	-	-	3/-	1.5/-
			Total	**14/11**	-	**12/12**	**20/17**
Second Semester							
1.		Language Elective	10,12	2	-	-	2
2.	21MAX02	Engineering Mathematics II	1	3	-	-	3
3.	21CYX01/ 21PYX01	Engineering Chemistry /Engineering Physics	1/1	3/3	-	-	3/3
4.	21BEX01/ 21BEX06	Basics of Engineering/ IT Workshop	1,12/1,12	-/3	-	3/-	1.5/3
5.	21BEX07	Python Programming	1,12	3	-	-	3
6.	21BEX08	Python Programming Lab	4	-	-	3	1.5
7.	21BEX05/ 21BEX04	Engineering Workshop / Engineering Drawing	1,9,10/1,5,10	-	-	3/3	1.5/1.5
8.	21CYX02/ 21PYX02	Engineering Chemistry Lab/Engineering Physics Lab	4/4	-	-	3/3	1.5/1.5
9.	-/21HSX02	-/Communicative English Lab	-/10,12	-	-	-/3	-/1.5
			Total	**11/14**	-	**12/12**	**17/20**
Third Semester							
1	21MA304	Probability and Statistics using Python	1,4, 12	3	-	2	4
2	21ML302	Artificial Intelligence	1,2,3	3	-	-	3
3	21CS303	Data Structures	1,2,12	3	-	-	3
4	21CS304	Digital Logic Design	1, 4	3	-	2	4
5	21DS305	Mathematical Foundation for Computer Science and Data Science	1,12,PSO1	3	-	-	3
6	21CS306	Object Oriented Programming with JAVA	1,2,3	3	-	-	3
7	21CS307	Data Structures Lab	2,3,4,5	-	-	3	1.5
8	21CS308	JAVA Lab	2,3,4,5	-	-	3	1.5
9	21BEA01	Environmental Studies	1,7	-	-	-	-
10	21ESX01	Employability Skills I	1,2,5,8,10, 12	0	-	2	-
11	21HSX11	CC & EC Activities I	6,7, 9,10	-	-	1	-
			Total	**18**	-	**13**	**23**

Fourth Semester							
1	21IT304	Database Management Systems	1,4,12	3	-	-	3
2	21IT403	Operating Systems	1,12	3	-	-	3
3	21CS403	Computer Organization and	1,12	3	-	-	3

		Architecture					
4	21CS404	Design and Analysis of Algorithms	2,3	3	-	2	4
5	21DS405	Foundations of Data Science	2, 3, PSO1, PSO2	3	-	-	3
6	21IT308	Database Management Systems Lab	4	-	-	3	1.5
7	21DS407	Foundations of Data Science Lab	4,5,8	-	-	3	1.5
8	21ESX01	Employability Skills I	1,2,5,8,10, 12	0	-	2	2
9	21HSX11	CC & EC Activities I	6,7, 9,10	-	-	1	1
			Total	**15**	**-**	**11**	**22**
Fifth Semester							
1	21IT405	Web Technologies (Integrated)	3,5,PSO1	3	-	2	4
2	21DS502	Deep Learning for Data Science	1,2,4,5,12	3	-	-	3
3	21DS503	Data Analytics &Visualization Techniques (Integrated)	1,4,PSO1	3	-	2	4
4	21ML504	Computer Networks	1, 2,3	3	-	-	3
5		Elective I (Professional Elective)		3	-	-	3
6		Elective II (Open Elective I)		3	-	-	3
7	21DS507	Deep Learning Lab		-	-	3	1.5
8	21TPX01	Term Paper	1,4,10,12	-	-	3	1.5
9	21ESX02	Employability Skills II	1,2,5,8,10,12	0	-	2	-
10	21HSX12	CC & EC Activities II	6,7, 9,10	-	-	1	-
11	21SIX01	Summer Internship I	1,2,8,10,12	-	-	-	1
			Total	**18**	**-**	**12**	**24**
Six Semester							
1.	21DS601	Optimization Techniques for ML	2,3,PSO1,PSO2	3	-	-	3
2.	21IT602	Automata and Compiler Design	1,2,3	3	-	-	3
3.	21CS603	Software Engineering	4,5,8,11, PSO1	3	-	-	3
4.		Elective III (Professional Elective)		3	-	2	4
5.		Elective IV (Open Elective II)		3	-	-	3
6.	21DS606	Optimization Techniques for ML Lab	4, 5	-	-	3	1.5
7.	21MPX01	Mini Project	4, 5,08	-	-	3	1.5
8.	21ESX02	Employability Skills II	All Pos & PSOs	0	-	2	2
9.	21HSX12	CC & EC Activities II	1,2,3,5,6,8,10,12	-	-	1	1
10.	21ATX01	Environmental Studies	6,7,9,10	-	-	-	-
11.	21ATX02	Professional Ethics and Human Values	1,7	-	-	-	-
12.	21ATX---	Audit Course	**----**	-	-	-	-
			12	**15**	**-**	**8**	**22**
Seventh Semester							
1		Elective V (Professional Elective)		3	-	-	3
2		Elective VI (Professional Elective)		3	-	-	3
3		Elective VII (Open Elective III)		3	-	-	3
4		Summer Internship II	1,2,5,6,10,12	-	-	-	1
5		Project	All POs& PSOs	-	-	16	8
			Total	**9**	**-**	**16**	**18**
Eighth Semester							
1		Elective VIII (Professional Elective)		-	-	-	3
2		Elective IX (Open Elective IV)		-	-	-	3
3	21FIX01	Full Semester Internship (FSI)	1,2,5,8,9,10, PSO1,PSO2	-	-	-	8
			Total	**6**	**-**	**-**	**14**

List of Electives

Language Electives							
No.	**Course Code**	**Course**	**POs**	**Contact Hours**			
				L	**T**	**P**	**C**
1	21HSX03	Advanced Communicative English	10,12	2	-	-	2
2	21HSX04	Communicative German		2	-	-	2
3	21HSX05	Communicative French		2	-	-	2
4	21HSX06	Communicative Japanese		2	-	-	2
5	21HSX07	Communicative Spanish		2	-	-	2
6	21HSX08	Communicative Korean		2	-	-	2
7	21HSX09	Communicative Hindi		2	-	-	2

Elective I							
Career Path I, II, III							
1	**21MLC11**	**Computer Vision & Pattern Recognition**	1,3,PSO1,PSO2	3	-	-	3
2	**21CSC21**	**Web Programming Languages (Full Stack Developer)**	1,2,7,12	3	-	-	3
3	**21MLC31**	**Fundamentals of Cloud Computing**	2,6,7,8	3	-	-	3
Non-Career Path (Core Electives)							
4	21CS004	Principles of Programming Languages	1, 2, 3, 4	3	-	-	3
5	21CS005	Mobile Computing	3, 8	3	-	-	3
6	21CS006	Distributed Operating Systems	1,2	3	-	-	3
7		MOOCs/Honors		3	-	-	3
Elective II: Open Elective I							
1	21CE001	Disaster Management	2, 7	3	-	-	3
2	21EE001	Electrical Installation and Safety Measures	2,3,6,8	3	-	-	3
3	21DS001	Fundamentals of Data Science	3,PSO1,PSO2	3	-	-	3
4	21ME001	Fundamentals of Optimization Techniques	1, 2	3	-	-	3
5	21EC001	Sensors for Engineering Applications	1, 2	3	-	-	3
6	21CS001	Fundamentals of Artificial Intelligence **(Except CSE & IT)**	1, 2, 3	3	-	-	3
7	21IT001	Fundamentals of Multimedia	1, 5, 7	3	-	-	3
Elective III							
Career Path I, II, III							
1	**21MLC12**	**Machine Learning for Business Intelligence**	2,3,PSO1,PSO2	3	-	2	4
2	**21CSC22**	**Web Application Developments Framework (Full Stack Developer)**	1, 3,4	3	-	2	4
3	**21MLC32**	**Cloud Services using AWS**		3	-	2	4
Non-Career Path (Core Electives)							
4	21CS007	Cloud Computing Essentials	2,5,6,7,8	3	-	2	4
5	21CS008	Cryptography and Network Security	3, 6,8	3	-	2	4
Elective IV: Open Elective II							
1	21CE002	Air Pollution and Environmental Impact Assessment	6, 7,12	3	-	-	3
2	21EE002	Renewable Energy Sources	2, 7	3	-	-	3
3	21ME002	Principles of Entrepreneurship	1,11	3	-	-	3
4	21EC002	Electronics for Agriculture	1, 2	3	-	-	3
5	21CS002	Fundamentals of Machine Learning	2, 3	3	-	-	3
6	21CH002	Industrial Safety and Hazard Management	2, 6, 7,8	3	-	-	3
7	21IT002	Fundamentals of Cloud Computing	1, 7	3	-	-	3
8	21BS002	Advanced Numerical Techniques	1, 2	3	-	-	3
9	21BS003	Functional Materials and Applications	1, 2	3	-	-	3
Elective V							
Career Path I, II, III							
1	**21MLC13**	**Conversational AI**	1,2,4,12,PSO1	3	-	-	3
2	**21CSC23**	**Web Application Databases (Full Stack Developer)**	2,3	3	-	-	3
3	**21MLC33**	**Cloud Security Essentials**	2,3	3	-	-	3

Non-Career Path (Core Electives)

4	21IT008	Social Network Analysis	2, 4, 5	3	-	-	3
5	21CS011	Optimization Techniques	2, 3, 5	3	-	-	3
6	21CS012	Wireless Adhoc Networks	2, 3	3	-	-	3
7		MOOCs/Honors		3	-	-	3

Elective VI

1	21CS014	Green Computing	2,3,6,7	3	-	-	3
2	21CS015	Software Project Management	3,6	3	-	-	3
3	21CS016	Soft Computing	1,2,3,4	3	-	-	3
4		MOOCs/Honors		3	-	-	3

Elective VII: Open Elective III

1	21CE003	Solid Waste Management	2,3,12	3	-	-	3
2	21EE003	Fundamentals of Electrical Vehicle Technology	2,3,12	3	-	-	3
3	21ME003	Industrial Engineering and Management	1,11	3	-	-	3
4	21EC003	Interfacing and Programming with Arduino	1,2	3	-	-	3
5	21CS003	Data Science for Engineering Applications	2,3,4	3	-	-	3
6	21CH003	Industrial Ecology for Sustainable Development	2,6,7	3	-	-	3
7	21IT003	Fundamentals of Mobile Computing	1,7	3	-	-	3
8	21BS004	Advanced Materials of Renewable Energy	1,7	3	-	-	3
9	21BS005	Applied Linear Algebra for Engineers	1,12	3	-	-	3

Elective VIII: Professional Elective

1	21CS017	Fundamentals of Social Network Analysis	2, 4,5	-	-	-	3
2	21CS018	Information Retrieval Systems	1,2,3,4	-	-	-	3
3	21CS019	Fundamentals of Devops	1,3, 5,8,10	-	-	-	3
4		MOOCs/Honors		-	-	-	3

Elective IX: Open Elective IV

1	21CE019	Green Buildings	2,3,4,5,7	-	-	-	3
2	21EE017	Sustainable Energy	1,2,12	-	-	-	3
3	21ME004	Total Quality Management	1,11	-	-	-	3
4	21EC011	Communication Technologies	1,2	-	-	-	3
5	21CS020	Applications of Artificial Intelligence	2,3,6,7	-	-	-	3
6	21CH016	Green Technologies	1,6,7	-	-	-	3
7	21IT015	Human Computer Interaction	1,7	-	-	-	3
8	21BS006	Handling of Industrial Waste and Wastewater	1,7	-	-	-	3

Audit Course

1	21AT001	Communication Etiquette in Workplaces	-	-	-	-	-
2	21AT002	Contemporary India: Economy, Policy andSociety	-	-	-	-	-
3	21AT003	Design The Thinking	-	-	-	-	-
4	21AT004	Ethics and Integrity	-	-	-	-	-
5	21AT005	Indian Heritage and Culture	-	-	-	-	-
6	21AT007	Intellectual Property Rights and Patents	-	-	-	-	-
7	21AT008	Introduction to Journalism	-	-	-	-	-
8	21AT009	Mass Media Communication	-	-	-	-	-
9	21AT010	Science, Technology and Development	-	-	-	-	-
10	21AT011	Social Responsibility	-	-	-	-	-
11	21AT012	The Art of Photography and Film Making	-	-	-	-	-
12	21AT013	Gender Equality for Sustainability	-	-	-	-	-
13	21AT014	Women in Leadership	-	-	-	-	-
14	21AT015	Introduction to Research Methodology	-	-	-	-	-
15	21AT016	Climate Change and Circular Economy	-	-	-	-	-

B. Tech. (Honors)

Domain I (Data Engineering)							
01	21CSH11	Advanced Data Structures	2,3,4	4	-	-	4
02	21CSH12	Advanced Databases	2,3,4	4	-	-	4

03	21CSH13	Programming, Data Structures and Algorithms Using Python	2,3,4,5	4	-	-	4
04	21CSH14	Bioinformatics	2,3	4	-	-	4
Domain II (Modern Software Engineering)							
01	21CSH21	DevOps	1,3,5,8,10	4	-	-	4
02	21CSH22	Design Patterns	2,3	4	-	-	4
03	21CSH23	Advanced Software Engineering	1,3,4, PSO1	4	-	-	4
04	21CSH24	Robotic Process Automation	3,5, 8, PSO2	4	-	-	4
Domain III (Security)							
01	21CSH31	Fundamentals of Systems Security	1,2	4	-	-	4
02	21CSH32	Python Programming for Security	2,3,4	4	-	-	4
03	21CSH33	Management of Information Security	3,6,7	4	-	-	4
04	21CSH34	Computer Forensics	2,3	4	-	-	4
Domain IV (User Interface Design)							
01	21CSH41	Computer Graphics	1,2,3,4	4	-	-	4
02	21CSH42	Multimedia Systems	3,4	4	-	-	4
03	21CSH43	Human Computer Interaction	2,3	4	-	-	4
04	21CSH44	Mobile Programming	3,4	4	-	-	4

B. Tech. (Minors)							
Energy Science & Technology							
01	21CHM11	Foundation of Energy Science and Technology	1,2,3,5,7,12	4	-	-	4
02	21CHM12	Energy Generation from Waste	1,2,3,4,5	4	-	-	4
03	21CHM13	Energy Storage Systems	1,2,3,6,7	4	-	-	4
04	21CHM14	Hydrogen Energy and Fuel Cells	1,2,3,7	4	-	-	4
Nano Science & Technology							
01	21CHM21	Introduction and Characterization of Nano Materials	1,2,3,7	4	-	-	4
02	21CHM22	Carbon Nanostructures and Applications	1,3,4,5	4	-	-	4
03	21CHM23	Energy, Environment & Biomedical Nanotechnology	1,2,3,7	4	-	-	4
04	21CHM24	Industrial Applications of Nano Technology	2,3,5,,7	4	-	-	4
Environmental Engineering							
01	21CEM11	Watershed Management	6,7	4	-	-	4
02	21CEM12	Industrial Pollution Control and Engineering	3,6,7,12	4	-	-	4
03	21CEM13	Solid and Hazardous Waste Management	1,3,6,7	4	-	-	4
04	21CEM14	Ecology and Environmental Assessment	1,3,6,7	4	-	-	4
Artificial Intelligence & Machine Learning							
01	21CSM11	Fundamentals of AI & Machine Learning	1,12	4	-	-	4
02	21CSM12	Feature Engineering for Machine Learning	1,2,3	4	-	-	4
03	21CSM13	Exploratory Data Analytics	1,4	4	-	-	4
04	21CSM14	Foundations of Deep Learning	1,2, 4	4	-	-	4
Cyber Security							
01	21CSM21	Fundamentals of Security	1,2	4	-	-	4
02	21CSM22	Management of Information Security	3,6,7	4	-	-	4
03	21CSM23	Cyber Security	1,3,4	4	-	-	4
04	21CSM24	Fundamentals of Cloud Security	2,3	4	-	-	4
Data Science & Analytics							
01	21CSM31	Data Cleaning	2,3,4	4	-	-	4
02	21CSM32	Data Engineering	1,2,3,4	4	-	-	4
03	21CSM33	Text Analytics	1,2,4	4	-	-	4
04	21CSM34	Social Network and Semantic Analysis	2, 4	4	-	-	4
Computer Systems Programming							
01	21CSM41	Programming Fundamentals	1,2,3	4	-	-	4
02	21CSM41	Data Structures & Algorithms	1,2,3,4	4	-	-	4
03	21CSM41	Fundamentals of Databases	1,4	4	-	-	4
04	21CSM41	Fundamentals of Computer Networks & Operating Systems	1,2,3	4	-	-	4

Digital IC Design							
01	21ECM11	Fundamentals of VLSI Design	1,2,3	4	-	-	4
02	21ECM12	Digital Design using HDL	1,2,3	4	-	-	4
03	21ECM13	FPGA Technology	1,2	4	-	-	4
04	21ECM14	Analog and Mixed Signal Design	1,2	4	-	-	4
Industrial Automation							
01	21ECM21	Microcontrollers and Interfacing	1,2,3	4	-	-	4
02	21ECM22	Sensors and Data Acquisition System	1,2	4	-	-	4
03	21ECM23	Fundamentals of Labview	1,2	4	-	-	4
04	21ECM24	Medical Robotics	1,2,3	4	-	-	4
Communicationsand Networking							
01	21ECM31	Principles of Communications	1,2	4	-	-	4
02	21ECM32	Coding Theory and Practice	1,2	4	-	-	4
03	21ECM33	Ad-hoc and Wireless Sensor Networks	1,2,3	4	-	-	4
04	21ECM34	Fundamentals of Multimedia Networking	1,2,3	4	-	-	4
Avionics							
01	21ECM41	Principles of Aerodynamics	1,2	4	-	-	4
02	21ECM42	Aircraft Electrical Systems	1,2	4	-	-	4
03	21ECM43	Aircraft Instrument Systems	1,2	4	-	-	4
04	21ECM44	Aircraft Communication and Navigational Systems	1,2	4	-	-	4
Geographic Information System							
01	21ECM51	Sensors and Sensing Technology	1,2	4	-	-	4
02	21ECM52	Geographic Information Systems	1,2	4	-	-	4
03	21ECM53	Digital Image Processing	1,2	4	-	-	4
04	21ECM54	Lidar Systems	1,2	4	-	-	4
Electric Vehicles Technology							
01	21EEM11	Introduction to Electric Vehicles Technologies	2,3	4	-	-	4
02	21EEM12	Electrical Drives and Controllers for Electric Vehicles	2,3	4	-	-	4
03	21EEM13	Charging Technology in Electric Vehicles	2,3	4	-	-	4
04	21EEM14	Computer Vision in Electric Vehicles	2,3	4	-	-	4
Electric Vehicles Technology							
01	21EEM21	Fundamentals of Smart City	2,3	4	-	-	4
02	21EEM22	Smart City Infrastructure	2,3	4	-	-	4
03	21EEM23	Computational Methods for Smart City Management	2,3	4	-	-	4
04	21EEM24	Communication Technologies and Mobility for Smart City	2,3	4	-	-	4
Electric Vehicles Technology							
01	21EEM31	Modelling and Simulations of Industrial Applications	2,3	4	-	-	4
02	21EEM32	Industrial Sensors and Actuators	2,3	4	-	-	4
03	21EEM33	Programmable Logic Controllers	2,3	4	-	-	4
04	21EEM34	Control Design for Industrial Applications	2,3	4	-	-	4
Cloud Application Development							
01	21ITM11	Introduction to Cloud Computing	6, 7, 12	4	-	-	4
02	21ITM12	Introduction to Web Development with HTML, CSS, JavaScript	1, 2, 3, 9, 12	4	-	-	4
03	21ITM13	Developing Cloud Native Applications	5, 8, 10	4	-	-	4
04	21ITM14	Developing Cloud Apps with Node.js and React	5, 8, 10	4	-	-	4
Robotics and Automation							
01	21MEM11	Introduction to Robotics	1,2,3	4	-	-	4
02	21MEM12	Drives and Sensors	1,2,3,4	4	-	-	4
03	21MEM13	Control Systems for Robotics	1,2,3,4	4	-	-	4
04	21MEM14	Machine Learning for Robotics	2,5	4	-	-	4
Industrial Systems Engineering							
01	21MEM21	Industrial Management	1,10,11,12	4	-	-	4
02	21MEM22	Fundamentals of Operations Research	1,2,3,5	4	-	-	4
03	21MEM23	Enterprise Resource Planning	1,2,3,5,11,12	4	-	-	4
04	21MEM24	Production Planning and Control	1,2,3,5,11,12	4	-	-	4

3rd Semester

21MA304 - Probability and Statistics using Python

3 0 2 4

Course Outcomes:

1. List the basic data structures of Python and various libraries of Python.
2. Illustrate basic probability axioms and apply Bayes' theorem related to engineering problems.
3. Differentiate between various distributions of random variables on given data.
4. Examine given dataset based on descriptive statistics.
5. Compare the null or alternative hypotheses using the suitable test statistic.
6. Choose appropriate curve fitting and correlation methods for a given data

CO-PO Mapping

CO	PO1	PO4	PO12
1	3	1	2
2	3	2	2
3	1	3	1
4	3	2	1
5	3	2	1
6	2	3	1

3-Strongly linked | 2-Moderately linked | 1-Weaklylinked

Unit I **12+8 Hours**

Fundamental of Python Programming

Strings, Files, Lists, Dictionaries, Tuples**. Introduction to NumPy:** Creating Numpy Arrays, Numpy Data Objects, dtype, Numerical Operations. Numpy Arrays: Concatenating, Flatten, reshape, adding Dimensions, Vector Stacking,Pandas DataFrame, Accessing and Changing values of DataFrames, Pandas Pivot, groupby **Introduction to Pandas:** Reading and Writing Data in Pandas, Multi-level Indexing, Data Visualization in Pandas

Objects and Classes

Practical Components

1. Experiments on Basic experiments on fundamentals on python
2. Exploratory Data Analysis using NumPy
3. Exploratory Data Analysis using Pandas
4. Exploratory Data Analysis using Matplotlib

Unit II **12+6 Hours**

Random Variables and Probability Distributions

Introduction to Probability: Events, Axioms of Probability and Probabilities of events; Conditional Probability, Bayes Theorem. Discrete and Continuous random variables; Expectation and Variance, Distribution of a Random Variables: **Discrete and Continuous Distributions-** Bernoulli, Binomial, Poison and Normal Distributions (without proofs); Central Limit Theorem (without proof).

Gaussian distribution

Practical Components

1. Simulation of random experiments and plot the empirical probability of each event, alongside the theoretical probability.
2. Python script to find conditional probability of a given event.
3. Simulation of various probability distributions

Unit III **12+10 Hours**

Sampling Theory and Testing of Hypothesis

Sampling: Population and sample, parameter and statistic, standard error. Parameter estimations, Confidence Intervals, Unknown Standard Deviation. **Hypothesis Testing**: Null and alternative hypothesis, level of significance, Type I and Type II errors, one tail and two-tail tests. Hypothesis testing concerning one mean and proportions, two means –Proportions and their differences using Z-test, Student's t-test, F-test, Chi-square test.

One-way ANOVA

Practical Components

1. Python Modules for Descriptive Statistics
2. Write a python program to perform z-test, t- test for testing the Null hypothesis for single mean
3. Write a python program to perform z-test, t- test for testing the Null hypothesis for difference of means
4. Write a python program to explore problems in F-test.
5. Write a python program to perform chi square for determine whether there is a significant association between the two variables

Unit IV **12+8 Hours**

Curve Fitting, Correlation & Regression

Curve Fitting: Introduction-fitting a straight line-second degree curve-exponential curve by method of least squares. **Correlation:** Simple correlation, Karl Pearson's Correlation coefficient and Spearman's rank correlation. Simple regression, multiple regression.

Covariance

Practical Components

1. Develop a straight line curve to fit some real world observations
2. Develop a exponential curve to fit some real world observations
3. Explore problems in Correlation with sample data
4. Explore problems in Regression with sample data

Total: 48+32 Hours

Text Books:

1. Probability & Statistics by T.K.V.Iyengar&B.Krishna Gandhi & Others, S.Chand (2016) 6th edition.
2. E. Walpole, R. H. Mayers, S. L. Mayers and K. Ye, (2012), Probability and Statistics for Engineers and Scientists,9th Edition, Pearson Education, ISBN: 978-8-131-71552-9.
3. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition, O'Reilly Media, Inc.

Reference Books:

1. Michael Baron, Probability and Statistics For Computer Scientists, 2nd Edition, CRC Press
2. Sheldon M. Ross, (2011), Introduction to Probability and Statistics for Engineers and Scientists, 4th Edition, Academic Foundation, ISBN: 978-8-190-93568-5.

Internal Assessment Pattern

Cognitive Level	**Int. Test 1 (%)**	**Int. Test 2 (%)**	**Assignment Test(%)**
Remember	10	10	--
Understand	50	30	--
Apply	40	60	--
Analyze	--	--	--
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	--

SAMPLE QUESTION (S)

Remember

1. Define variable in python.
2. List any two differences between built-in and user defined functions.
3. List any two python packages used mathematics.

Understand

1. Explain use of probability and statistics in real world.
2. Illustrate about random variables in python.

Apply

1. Find different real time applications which use Markov process
2. Implement a queuing theory to handle multiple processes.

21ML302 Artificial Intelligence

3 0 0 3

Course Outcomes

1. Summarize the fundamental concepts of Artificial Intelligence.
2. Illustrate the Concepts of Heuristic Search Techniques.
3. Demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information
4. Make use of concept of Game Playing Algorithms.
5. Outline the concept of Planning System
6. Explain the Natural language processing system

CO–PO Mapping

CO	PO1	PO2	PO3
1	2	1	2
2	2	2	3
3	3	1	1
4	1	2	2
5	1	3	2
6	1	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I 12 Hours

Introduction

Definition, AI problems, AI techniques, Defining problem as a state space search, Production systems-Control Strategies, Problem characteristics, Production system characteristics. Problems – tic-tac-toe, 8-puzzle problem, Chess problem, Water Jug Problem, Missionaries and cannibals problem, Monkey and banana problem.

Application of AI-Tower of Hanoi problem- *Travelling Salesman Problem*

Unit II 12 Hours

Heuristic Search Techniques

Generate-and-test, Hill climbing, Best-first-search – OR Graphs – A* Algorithm, Problem reduction – AND-ORGraphs – AO* Algorithm, constraint satisfaction - cryptarithmetic problem.

Depth-first search – Breadth-first search- Data abstraction

Unit III 12 Hours

Knowledge representation

Knowledge Representation Issues, Representation and mapping, Approaches to Knowledge Representation, Frame Problem, Propositional logic, Predicate logic, Resolution, weak slot and filler structure-semantic nets.

Filler structures-conceptual dependency-scripts

Unit IV 12 Hours

Game Playing and Planning

Mini-max search, Alpha-beta cutoffs, planning system, Block world problem, goal stack planning, hierarchical planning, Natural language processing, syntactic processing. Decision trees, Perception, Vision, Speech recognition.

Learning - Navigation – Manipulation - Robot Architecture

Textbook (s)

1. E. Rich K .Knight, and B. Nair, Artificial Intelligence, 3rdEdition, TMH, 1 July 2017
2. Russel Norvig, Artificial Intelligence A modern Approach, 3rd Edition, Pearson Education, 2010

Reference (s)

1. Patrick henry Winston, Artificial Intelligence, third edition, Pearson Education Asia, 2005
2. Dan W. Patterson, Introduction to Artificial intelligence and Expert Systems, 2nd Edition, PHI, 2009

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	40	40	--
Understand	30	40	--
Apply	30	20	70
Analyze	--	--	30
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. State the tasks, which are associated with A.I.
2. Give an example of crypt-arithmetic problem.
3. Define Speech Recogonition.
4. Recall Intelligence

Understand

1. Explain Turing Test.
2. Explain the characteristics of production system.
3. Write A* algorithm in detail and explain its functionality to solve 8-puzzle problem.
4. Explain the semantic analysis phase done through case grammars in Natural Language understanding

Apply

1. Distinguish between weak and strong slot filler structures
2. Discuss the tic-tac-toe problem in detail and explain how it can be solved using AI technique
3. Construct semantic net representation for the following:
   (i) Pompeian (Marcus), Blacksmith (Marcus)
   (ii) Mary gave the green flowered vase to her favorite cousin.

Analyze

1. Show that the tower of Hanoi problem can be classified under the area of AI. Give a state space representation of the problem.
2. Give an example of a problem for which breadth-first search would work better than depth-first search. Justify your answer.
3. Consider the following sentences:

   John likes all kinds of food.
   Apples are food.
   Chicken is food
   Anything anyone eats and isn't killed by food.
   Bill eats peanuts and is still alive.
   Sue eats everything Bill eats.

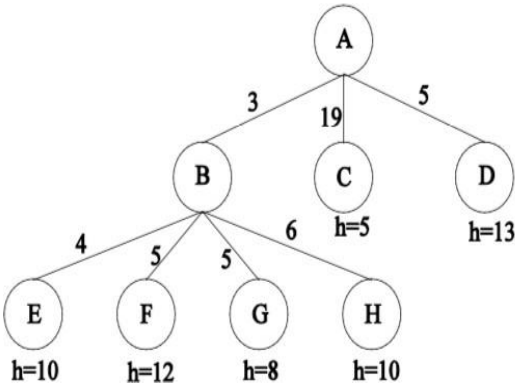
   (a) Translate these sentences into formula in predicate logic.
   (b) Convert the formulas of part a into clause form.

Prove that John likes peanuts using resolution

Open Book Exam Questions

Question 1:

Consider the following search tree produced after expanding node A, where each arc is labeled and the leaves are labeled with the value of a heuristic function, h.



(i) Identify the node, which is expanded next by each of the following search methods?

(a) Breadth First Search

(b) Depth-First Search

(c) Best-First Search

(d) A* Search

(e) AO* Search

(ii) Construct the Min Max Tree for the above figure (ignoring the cost values)

(iii) Identify whether the tree is eligible for pruning or not, if it is yes trace the alpha beta prunes

21CS303 Data Structures

3 0 0 3

Course Outcomes

At the end of the course, students will be able to

1. Describe the operations and implementation of List ADT
2. Comprehend the operations and implementation of Stack and Queue
3. Illustrate the applications of linear data structures
4. Describe the operations and implementation of hash table
5. Comprehend the operations and implementation of tree data structure
6. Illustrate the variations of tree data structure

CO-PO Mapping

CO	PO1	PO2	PO12
1	3	2	1
2	3	2	1
3	2	3	1
4	2	3	1
5	3	2	1
6	2	3	1

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I

Linear Data Structures–List **12 Hours**

AbstractDataTypes(ADTs)–ListADT–Array-basedimplementation–Linkedlistimplementation–Singly linked lists- Circularly linked lists- Doubly-linked lists – Applications of lists –Polynomial Manipulation – All operations (Insertion, Deletion, Merge, Traversal) - Searching - Linear Search – BinarySearch

Doubly linked Circular list

Unit II

Linear Data Structures – Stack and Queue **12 Hours**

Stack ADT – Array implementation – Linked list implementation – Applications of Stack – infix to postfix conversion, evaluation of postfix expression – Queue ADT – Array implementation – Linked list implementation – Application of Queue – Ticketcounter

Circular Queue

Unit III

Sorting, Hashing Techniques and Trees **12 Hours**

Sorting – Bubble Sort – Selection Sort – Insertion Sort – Shell Sort – Radix Sort – Quick Sort – Merge Sort - Hashing - Hash Functions – Separate Chaining – Open Addressing –Rehashing
Trees: Introduction, Terminology, Binary Trees, Representation of Binary Trees using arrays and linked lists, Binary tree traversals

Extendible Hashing

Unit IV

Variations on Trees and Graphs **12 Hours**

Binary Search Trees: definition, basic operations of BST (Searching, Insertion and deletion) - Introduction to AVL trees: Balancing AVL tree by rotations after insertions and deletions of a data node Multi-way search trees: Introduction to m-way search trees, B-trees, B+Trees;
Heaps: Binary heaps, definition of a Max-heap, Min-heap, Creating Max-Heap, Applications: Heap sort
Graphs: Terminology, Representation, Traversals: Depth First Search and Breadth First Search

Priority queue operations: insertions and extract-max

Total: 48 Hours

Textbook (s)

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Pearson Education,2002
2. Michael Main, Walter Savitch, Data Structures and other objects using C++, 4thEdition, Addison Wesley,2018

Reference (s)

1. S. Tanenbaum, Y. Langsam and M.J. Augenstein,, Data Structures using C and C++, 2ndEdition, Pearson Education,2015
2. R. F. Gilberg, B. A. Forouzan, Data Structures A Pseudocode Approach with C, 2nd Edition, CENGAGE Learning,2005

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Examination[3] (%)
Remember	20	20	--
Understand	60	60	--
Apply	20	20	80
Analyze	--	--	20
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. What are abstract datatypes?
2. List any 2 disadvantages of array
3. Define linked list
4. Define data structure
5. List any 2 applications of queue

Understand

1. Compare linked list with array
2. Explain Bubble Sort Process with an example
3. Demonstrate with neat diagram and algorithm to insert a node before the given key
4. Explain Deletion process using an example binary search tree
5. Explain why the selection sort is more efficient than the bubble sort
6. Explain with suitable example of LL rotation after inserting a new node into an AVL tree
7. Demonstrate the application of singly linked lists for the addition of the polynomials P1 andP2

Apply

1. Develop an algorithm to concatenate two single linked lists
2. Construct a priority queue and implement all basic operations to demonstrate priority queue
3. Build a recursive procedure to count the number of nodes in a binary tree

Sample Questions for Open Book Examination

Apply

1. SelectappropriatedatastructuretosimulatetheoperationsofaMusicPlayer–Songsinmusic playerarelinkedtopreviousandnextsong.youcanplaysongseitherfromstartingorending of the list.

[3] Open book Examination should contain only questions related to Higher Order Thinking (HOT) Skills

2. A bracket is considered to be any one of the following characters: (,), {, }, [, or]. Two brackets are considered to be a matched pair if the an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets:[],{},and().A matching pair of brackets is not balanced if brackets it encloses are not matched. For example, {[(])} is not balanced because the contents in between { and } are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, (, and the pair of parentheses encloses a single, unbalanced closing square bracket,]. By this logic, we say a sequence of brackets is balanced if the following conditions are met: It contains no unmatched brackets. The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets. Given n strings of brackets, determine whether each sequence of brackets is balanced. If a string is balanced, return YES. Otherwise, return NO.
3. You are given a stack of **N** integers such that the first element represents the top of the stack and the last element represents the bottom of the stack. You need to pop at least one element from the stack. At any one moment, you can convert stack in to a queue. The bottom of the stack represents the front of the queue. You cannot convert the queue back into a stack. Your task is to remove exactly **K** elements such that the sum of the **K** removed elements is maximized.
4. Vikas is given a bag which consists of numbers (integers) blocks, Vikas has to organize the numbers again in the same order as he has inserted it in to the bag, i.e. the first number inserted into the bag by Vikas should be picked up first followed by other numbers in series. Help Vikas to complete this work in O(n) time complexity with the condition to use one extra bag to complete the work (assume that the bags are compact and is in the form of a stack structure and has the same width as that of the number blocks and is large enough to fill the bag to the top and the number taken from bag is in reverse order).

21CS304 Digital Logic Design

3 0 2 4

Course Outcomes

At the end of the course, students will be able to

1. Understand different number systems, its conversions, and binary arithmetic.
2. Classify logic circuits using basic Logic gates and simplify logic expressions using theorems, K-map.
3. Design and implement logical devices using combinational circuits.
4. Demonstrate and compare the construction of programmable logic devices and different types of ROM.
5. Understand and Analyze Sequential circuits like latches and flip-flops.
6. Analyze and Design sequential circuits like Registers and Counters.

CO-PO Mapping

CO	PO1	PO4
1	3	2
2	3	2
3	3	3
4	3	2
5	3	3
6	2	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12+6 Hours**

Number systems and Boolean algebra

Review of Number Systems, Conversion of Numbers from One Radix to Another Radix, Complements, Representation of Negative Numbers, Binary Arithmetic, Binary Codes, Error detecting & correcting codes, Basic Theorems and Properties of Boolean Algebra, Digital Logic Gates, Universal Gates.

IC specifications and pin diagram of gates

Practical Components

1. Realization of Logic gates and verification of Truth tables
2. Realization of basic gates using Universal gates (NOR)
3. Realization of basic gates using Universal gates (NAND)

Unit II **12+12Hours**

Boolean Function Minimization

Minimization of Switching Functions using K-Map up to 4-variables, Prime implicants, don’t care combinations, Minimal SOP and POS forms

Combinational Arithmetic Logic Circuits-1

Adders, Subtractors, Binary Multiplier-*parity bit Generator,* Ripple carry adder, Multiplexer, De-Multiplexer, Encoder, Priority encoder, Decoder, MUX Realization of switching functions.

Code Converters, Magnitude Comparator

Practical Components

1. Implementation of Half Adder
2. Implementation of Full Adder
3. Implementation and verification of Encoder
4. Implementation and verification of Decoder
5. Implementation and verification of MUX
6. Implementation and verification of DE-MUX

Unit III **12+6Hours**

Programmable Logic Devices and Sequential Logic Circuits-1

Basic PLD’s-ROM-PROM-PLA-PAL - Realization of Switching functions using PLD’s Classification of Sequential Circuits (Synchronous and Asynchronous): Latches and Basic Flip-Flops-Truth Tables and Excitation Tables

Conversion of flip-flops

Practical Components

1. Realization of Flip-Flops using ICs (SR Flip Flop)
2. Realization of Flip-Flops using ICs (JK Flip Flop)
3. Realization of Flip-Flops using ICs (D and T Flip Flop)

Unit IV **12+8 Hours**

Sequential Logic Circuits - II

Design of Registers - Buffer Register - Control Buffer Registers - Bidirectional Shift Registers - Universal Shift Register - Design of Synchronous Counters – Ripple counter, Up-down Counters, Design of Asynchronous Counters-Variable Modulus Counters (Mod-2,4,6,10 & 16), Ring Counter, Johnson Counter, Sequence generator.

Sequence detector

Practical Components

1. Verification of SISO and SIPO Shift Registers
2. Verification of PIPO and PISO Shift Registers
3. Implementation of Synchronous Counter
4. Implementation of Asynchronous Counter

Total: 48+32 Hours

Textbook (s)

1. Digital Design 6th Edition by M Morris Mano, PEARSONINDIA,2018.
2. Charles H. Roth, Fundamentals of Logic Design,3rdEdition, ThomsonPublications,2014
3. John F. Wakerly, Digital Design Principles & Practices, 4thEdition, PHI/ Pearson EducationAsia,2008

Reference (s)

1. ZviKohavi, Switching & Finite Automata theory, 2nd Edition,TMH,2008
2. R P Jain,Modern Digital Electronics, 3rd Edition,TMH,2003
3. A. Anand Kumar, Switching Theory and Logic Design, 3rd Edition,PHP,2016.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Assignment Test[4] (%)
Remember	20	10	--
Understand	50	40	--
Apply	30	30	--
Analyze	--	20	--
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	--

SAMPLE QUESTION (S)

Remember

1. Retrieve the decimal value of the fractional binary number0.1011
2. List any two postulates of Boolean algebra
3. List the four uses of Multiplexer.
4. Define Flip flop and Latch
5. Define Synchronous sequential circuit

Understand

1. Explain working functionality of Programable Logic Devices(PLD).
2. Represent the following Boolean expression to SOP and POS form: A+BC'+ABD'+ABCD
3. Represent the Boolean function T=F (w, x, y, z) = ∑ m (0,1,2,4,5,7,8,9,12,13) by using 8 to1mux.
4. Illustrate a half adder using NAND –NAND
5. Represent a T flip flop using JK flipflop.
6. Differentiate between Bi-directional and Universal Shift-Registers.

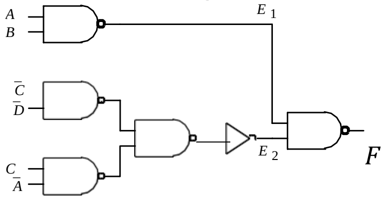
[4] Assignment test should contain only questions related to Higher Order Thinking (HOT) Skills

Apply

1. Find a circuit that has no static hazards and implements the Boolean function F(A,B,C,D) = Σ (0,2,6,7,8,,10,12)..
2. Construct a four input NAND gate using gates with 2 inputs. Write down the truth table. Do the same for a 4 input NOR gate.
3. Implement MOD-6 and Johnson Counter.
4. Construct a combinational circuit to convert BCD to EX-3code.
5. Implement a combinational logic circuit, which can compare two bits binary numbers.

Analyze

1. Analyze the following in two ways to get sums of product and product of sums equations. Then show how you can get one from the other using DeMorgan's Theorem



2. Why the input variables to a PAL are buffered
3. Why the fixed format in not used and floating format issued
4. What happens when an electric signal is grounded?
5. Design a Sequence detector to detect10111001

21DS305 Mathematical Foundation for Computer Science and Data Science

3 0 0 3

Course Outcomes

After undergoing this course, the learners are enabled to

1. Demonstrate Vector Spaces , Inner product spaces and their applications.
2. Solve Problems involving Recurrence Relations and Generating Functions
3. Apply Eigenvalues in Singular value decomposition and Principal component analysis.
4. Derive statements using the rules of inference for Mathematical logic.
5. Understand the basic knowledge of Graph theory and applications.
6. Construct the minimum spanning tree using different algorithms.

CO-PO Mapping

CO	PO$_1$
1	3
2	3
3	3
4	3
5	3
6	3

3-Strongly linked | 2-Moderately linked | 1-Weaklylinked

Unit-I

Vector Spaces and Inner Product Spaces **12 Hours**

Vector Spaces: Definition, Properties of Vector Spaces, Subspaces, Algebra of Subspaces, Basis and Dimension, Linear Span, Linear Independence and Dependence of vectors, Linear Transformations, Range and Null Spaces, Rank-Nullity theorem, Matrices of Transformations. (All theorems without Proof)

Inner Product Spaces: Definition, Properties of Inner Product Spaces, Norm of a vector, Cauchy Schwarz's Inequality, Complete orthonormal set, Gram-Schmidt Orthogonalization process.

Bessel's inequality

Unit-II

Recurrence Relations and Advanced Linear Algebra **12 Hours**

Recurrence Relations: Generating Functions – Calculating Coefficient of Generating Functions – Recurrence Relations – Solving Recurrence Relations by Substitution method, Solving Recurrence Relations by generating functions.

Advanced Linear Algebra: Hermitian, Skew-Hermitian, Unitary Matrices, Eigenvalues and Eigenvectors of complex matrices, Singular Value Decomposition (SVD), Principal Component Analysis (PCA)

Method of Characteristic roots

Unit-III

Mathematical Logic **12 Hours**

Propositional Logic: Statements and Notations – Connectives – Well-formed Formulas – Truth Tables – Tautology – Equivalence Implication – Normal Forms – Rules of Inference for Statement Logic – Proof Techniques – Proof by Contradiction – Method of Induction.

Predicative Logic: Quantifiers - Universal Quantifiers - Free & Bound Variables - Rules of Inference for Predicate Logic.

Relations in Databases

Unit-IV

Graph Theory **12 Hours**

Basics of Graph Theory: Representation of Graphs – Degree of a Graph –Complete Graphs – Regular Graphs – Bipartite Graphs –Walk – Path – Circuit – Cycle – Planar Graphs – Euler's Formula – Isomorphism – Euler Circuit and Hamilton Cycle – Chromatic Number.

Trees: Definition and properties of tree, Spanning Tree, Breadth First Search (BFS) algorithm, Depth First Search (DFS) algorithm, Minimum Spanning Tree, Kruskal algorithm, Prim's algorithms.

Applications of Graphs and Trees, Ternary Tree

Textbook(s)

1. Gilbert Strang, "Linear Algebra and Learning from Data", Wellesley, Cambridge Press, 2019.
2. Kenneth H. Rosen, Discrete Mathematics and Applications, 7th Edition, Tata McGraw Hill, 2015.
3. J. L. Mott, A. Kandel & T. P. Baker, Discrete Mathematics for Computer Scientists & Mathematicians, Prentice Hall India, 2nd Edition, 2010.

Reference (s)

1. Tremblay J.P. and P. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill,2007.
2. Erwin Kreyszig, Advanced Engineering Mathematics, Wiley India, 2015.
3. K Hoffman and R Kunze, Linear Algebra, Pearson Education, 2017.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Assignment Test/ Open Book Exam (%)
Remember	30	30	--
Understand	40	40	--
Apply	30	30	60
Analyze	--	--	40
Evaluate	--	--	
Create	--	--	--
Total (%)	100	100	100

Remember

1. Define Vector Space.
2. State Hand shaking theorem.

Understand

1. Prove that "if n^2 is odd then n is odd" by the method of contradiction.
2. Explain about Euler circuit and Hamiltonian cycle with suitable examples.

Apply

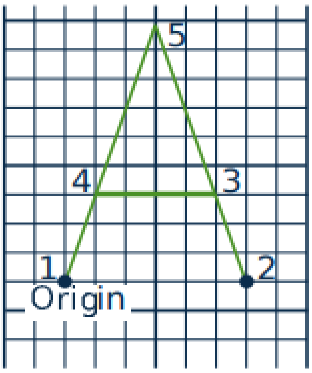
1. Find rank of the transformation $T: R^2 \rightarrow R^3$ defined by $T(x, y) = (x + y, x - y, y)$ using Rank-Nullity Theorem.
2. Obtain the matrices U, V and S such that $A = USV^T$, where $A = \begin{bmatrix} 3 & 3 & 2 \\ 2 & 3 & -2 \end{bmatrix}$ using Singular Value Decomposition.

Analyze

1. Prove the validity of the following argument: "If I study, then I will pass in exam. If I do not go to cinema, then I will study. But I failed in exam. Therefore, I went to cinema."
2. Draw a graph which contains an Eulerian circuit but not an Hamiltonian cycle with justification

Open Book Exam

1. Construct an AVL tree by inserting the following elements in the order given: {10, 6, 9, 4, 8, 7, 2, 11, 12, 3, 5, 1}, and balancing as necessary. Delete the elements in the following order {9, 6, 2, 5, 10, 3, 1, 4, 7, 8, 12, 11} balancing as necessary. Show the intermediate steps of deletion when re-balance (rotation) is used and end result of the insertion.
2. Model the matrix for the figure given below.
   i. Draw the image corresponding to the matrix obtained by multiplying $\begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}$ with obtained matrix in (i)
   ii. Sketch the image and write the corresponding matrix by rotating the following figure



through the angle $\frac{\pi}{6}$ about the point $\begin{bmatrix} 4 \\ 5 \end{bmatrix}$.

21CS306 Object Oriented Programming with JAVA

3 0 0 3

Course Outcomes

1. Summarize object oriented programming concepts
2. Develop applications using different types of inheritances
3. Create and use user defined packages
4. Analyze and recover runtime exceptions arise in the applications
5. Apply parallel processing applications using threads
6. Develop interactive applications for standalone and Internet

COs–POs Mapping

COs	PO1	PO2	PO3
1	3	2	3
2	3	1	3
3	1	1	2
4	2	1	3
5	3	1	3
6	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **13 Hours**

Introduction to Java

Overview of Object Oriented Programming principles, Importance of Java to the Internet, Byte code, Methods, classes and instances, Data types, arrays, control statements, simple java program, Classes and Objects–constructors, methods, access control, this keyword, overloading methods and constructors, garbage collection, java.lang – String, StringBuffer, StringBuilder, Type wrappers – Integer, Character, Float, Double, Boolean

Features of object oriented programming–Java History–Computer Programming Hierarchy–Role of Java Programmer in Industry

Unit II **11 Hours**

Inheritance, Packages & Interface

Inheritance: Basics, Using super, Multilevel Hierarchy, Method overriding, Dynamic Method Despatch, Using Abstract classes, Using final with inheritance, Sealed Classes

Packages: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, Member access rules, Java Collections – Arrays, ArrayList, Hashtable, StringTokenizer, Vector

Interface: Defining an interface, differences between classes and interfaces, implementing interface, variables in interface and extending interfaces, Default Methods in Interfaces, Static Methods in Interface

Nested–Inner Class & Anonymous Classes–Generic Class Types

Unit III **12 Hours**

Exception Handling & Multithreading

Exception handling: Concepts and benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built-in and User Defined Exceptions

Multithreading: Definition thread, thread life cycle, creating threads, synchronizing threads, daemon threads

Control Flow In Exceptions– JVM reaction to Exceptions– Inter Communication of Threads– Critical Factor in Thread–Deadlock

Unit IV **12 Hours**

Applets & Event Handling

Applets: Concepts of Applets, life cycle of an applet, types of applets, creating applets, passing parameters to applets, The AWT class hierarchy, user interface components labels, button, Text components

Event Handling: Events, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes, Compare basic AWT components with swing components, More user interface components-canvas, scrollbars, check box, choices, lists panels–scroll pane, dialogs, menu bar, layout managers.

Applet Security Policies–Anonymous Inner classes a Short–cut to Event Handling–Java Utilities (java.util Package)

Total: 48 Hours

Textbook (s)

1. H. Schildt, Java: The complete reference, 7th Edition, TMH, 2006
2. T. A. Budd, An Introduction to Object–Oriented Programming, 3rd Edition, Addison Wesley Longman, 2002

Reference (s)

1. Dietal&Dietal, Java: How to Program, 8th Edition, PHI, 2010
2. C. S. Horstmann and G. Cornell, Core Java, Vol 1. Fundamentals, 7th Edition, Pearson Education, 2004
3. C. Horstmann, BIG JAVA Compatible with Java 5 & 6, 3rd Edition, Wiley Publishers, 2008

SAMPLE QUESTION (S)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Assignment Test[1] (%)
Remember	25	15	10
Understand	30	15	30
Apply	25	40	30
Analyze	20	20	30
Evaluate	--	10	--
Create	--	--	--
Total (%)	100	100	100

Remember

1. List all OOPs principles
2. Define class and object
3. Illustrate various data types supported by JAVA
4. State various access controls available in JAVA
5. Define inheritance
6. Define thread

Understand

1. Explain byte code
2. Explain the use of super, static and final keywords
3. Illustrate the use of try, catch, throw and throws
4. Java is purely object oriented programming language. Explain
5. Explain types of polymorphism with Example

Apply

1. Write statements that perform the following one-dimensional-array operations:
   a) Set the 10 elements of integer array counts to zero
   b) Add one to each of the 15 elements of integer array bonus
   c) Display the five values of integer array best Scores in column format
2. Create two arrays that store all even no's in one array and all odd no's in another array for a given set of elements
3. Write a java program to implements final variable, Method and classes
4. Explain the reason for the following code giving compiling errors.

```
abstractclassAbstractClass
{
  abstractvoidabstractMethod()
  {
    System.out.println("First Method");
  }
}
```

5. Write a java program to implement the following techniques
   a) Method Overloading
   b) Method Overriding
6. Write a Java program to implement Virtual function technique

Analyze

1. Compare and Contrast an abstract class and an interface
2. Differentiate method overloading and overriding
3. Differentiate abstraction and encapsulation
4. Identify the uses of calling a synchronize method inside a synchronize method
5. Identify memory leak problems in Java objects

21CS307 Data Structures Lab

0 0 3 1.5

Course Outcomes

At the end of the course, students will be able to

1. Implement stack and queue data structures using array and linked list
2. Demonstrate the applications of stack and queue data structures
3. Implement sorting and searching algorithms and to compare their efficiency
4. Implement binary tree, binary search tree and tree traversals
5. Solve graph problems using appropriate data structure
6. Implement hashing techniques for real world applications (telephone directory, dictionary)

CO-PO Mapping

CO	PO2	PO3	PO4	PO5
1	3	2	3	2
2	3	2	3	1
3	3	1	2	2
4	2	2	2	2
5	2	2	3	2
6	2	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

List of Experiments

1. **Arrays**

Implement list using an array and perform the following operations on it.

- **Insert(pos, element):** It should insert the given element into the list at the specified position pos. If pos is out of range (less than 0 or greater than the current list size), display an error message.
- **Delete(pos, element):** This function should delete the given element from the list at the specified position pos, if it exists. If pos is out of range or the element is not found at the given position, print an appropriate message.
- **Search(element):** This function should check if the given element is present in the list. If the element is found, return True; otherwise, return False.
- **Running Sum of 1D Array()**: This function return a new array containing the running sum of input array 'num'. The running sum of an array is defined as
Running Sum[i] = sum(nums[0]…nums[i]).
- **Display():** This function should display the current elements present in the list.

Examples:

Initial list: []
Insert(0, 5): [5]
Insert(1, 10): [5, 10]
Insert(5, 15): Invalid position, element not inserted
Display: [5, 10]

Initial list: [1, 2, 3, 4]
Delete(1, 2): [1, 3, 4]
Delete(0, 5): Element not found at position 0
Display: [1, 3, 4]

List: [1, 2, 3, 4]
Search(3): True
Search(5): False

Given array: [1, 2, 3, 4]
Running Sum: [1, 3, 6, 10]

Given array: [2, 4, 6, 8]
Running Sum: [2, 6, 12, 20]

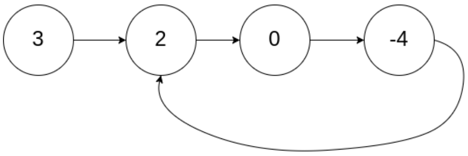
2. **Singly linked list**

Implement a singly linked list with the following functions: insert, delete, display and detect_cycle.

Node Structure:
Each node should have two attributes: val to store the value of the node and next to point to the next node in the list.

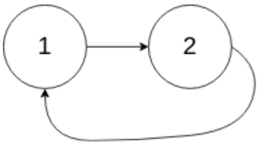
- **insert(val)**: This function should take an integer val as input and insert a new node with the given value at the end of the linked list.
- **delete(val)**: This function should take an integer val as input and remove the first occurrence of the specified value from the linked list. If the value is not found in the list, print an appropriate message.
- **display():** This function should display all the elements present in the linked list, starting from the head to the tail.
- **detect_cycle():** This function should detect if there is a cycle in the linked list. It should return True if a cycle is found and False otherwise. There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer.

Example 1: detect_cycle()



Input: head = [3,2,0,-4],
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).

Example 2: detect_cycle()



Input: head = [1,2]
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.
Example 3: detect_cycle()

Input: head = [1], pos = -1
Output: false
Explanation: There is no cycle in the linked list.

3. **Doubly linked list**

Implement a doubly linked list with the following functions: insert, delete, display, and middle_node.

Node Structure:
Each node should have three attributes: val to store the value of the node, prev to point to the previous node, and next to point to the next node in the list.

- **insert(val):** This function should take an integer val as input and insert a new node with the given value at the end of the doubly linked list.
- **delete(val):** This function should take an integer val as input and remove the first occurrence of the specified value from the doubly linked list. If the value is not found in the list, print an appropriate message.

- **display():** This function should display all the elements present in the doubly linked list, starting from the head to the tail.
- **middle_node():** This function should find and return the middle node of the doubly linked list. If the list has an even number of nodes, return the first middle node.

Example 1: middle_node(
Input: head = [1,2,3,4,5]
Output: [3,4,5]
Explanation: The middle node of the list is node 3.

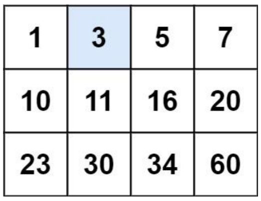
Example 2: middle_node(
Input: head = [1,2,3,4,5,6]
Output: [4,5,6]
Explanation: Since the list has two middle nodes with values 3 and 4, we return the first one.

4. **Linear search and Binary search**

Linear search
You are given an m x n integer matrix with the following two properties:

- Each row is sorted in non-decreasing order.
- The first integer of each row is greater than the last integer of the previous row.
- Given an integer target, return true if target is in matrix or false otherwise.

Example 1:



1	3	5	7
10	11	16	20
23	30	34	60

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3
Output: true

Binary Search:
Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

Example 1:

Input: nums = [3,0,1]
Output: 2
Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.
Example 2:

Input: nums = [0,1]
Output: 2
Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.
Example 3:

Input: nums = [9,6,4,2,3,5,7,0,1]
Output: 8
Explanation: n = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in nums.

Constraints:
All the numbers of nums are unique.

5. **Stack**

Remove All Adjacent Duplicates in String using stack

Implement a stack Abstract Data Type (ADT) that supports the following operations:

- push(element): Add the given element to the top of the stack.
- pop() -> element: Remove and return the top element from the stack. If the stack is empty, return None.
- peek() -> element: Return the top element of the stack without removing it. If the stack is empty, return None.

Implement a function that removes all adjacent duplicates in the string using a stack-based approach. A duplicate removal operation consists of selecting two adjacent and equal letters and eliminating them from the string. You should repeatedly apply these duplicate removals on the string s until no further duplicates can be removed.

- Input: A string s (1 ≤ |s| ≤ 10^5) consisting of lowercase English letters.
- Output: Return a string representing the final string after all adjacent duplicates have been removed.

Example 1:
Input: s = "abbaca"
Output: "ca"
Explanation:
For example, in "abbaca" we could remove "bb" since the letters are adjacent and equal, and this is the only possible move. The result of this move is that the string is "aaca", of which only "aa" is possible, so the final string is "ca".
Example 2:
Input: s = "azxxzy"
Output: "ay"

6. **Stack applications: Infix to postfix conversion**

Implement function to convert infix expression to its postfix notation using a stack.

Infix notation is the commonly used notation to represent mathematical expressions, where operators are written between the operands. For example, 2 + 3 * 4 is an infix expression.Postfix notation (also known as Reverse Polish Notation) is a way to represent mathematical expressions where each operator follows its operands. For example, the postfix equivalent of 2 + 3 * 4 is 2 3 4 * +.

- Input:A string infix_expr containing the infix expression. The expression will only contain the following characters: digits (0-9), operators (+, -, *, /), and parentheses (,).
- Output:A string representing the postfix expression obtained after converting the infix expression.

Example 1:
Input: infix_expr = "2 + 3 * 4"
Output: "2 3 4 * +"

7. **Stack applications: Evaluation of postfix expressions.**

Postfix expression as input and evaluates it to obtain the result using a stack.

- Input: A string postfix_expr containing the postfix expression. The expression will only contain the following characters: digits (0-9) and operators (+, -, *, /).
- Output: An integer representing the result of the evaluated postfix expression.

Examples:
Input: postfix_expr = "2 3 4 * +"
Output: 14
Explanation: The postfix expression is evaluated as 2 + (3 * 4) = 2 + 12 = 14.

8. **Queue**

Implement the queue Abstract Data Type (ADT) and use it to calculate the time needed to buy tickets.
Problem Statement: **Time Needed to Buy Tickets**

- There are n people in a line queuing to buy tickets, where the 0th person is at the front of the line and the (n - 1)th person is at the back of the line.
- You are given a 0-indexed integer array tickets of length n where the number of tickets that the ith person would like to buy is tickets[i].
- Each person takes exactly 1 second to buy a ticket. A person can only buy 1 ticket at a

time and has to go back to the end of the line (which happens instantaneously) in order to buy more tickets. If a person does not have any tickets left to buy, the person will leave the line.

- Return the time taken for the person at position k (0-indexed) to finish buying tickets.

Example 1:
Input: tickets = [2,3,2], k = 2
Output: 6
Explanation:
- In the first pass, everyone in the line buys a ticket and the line becomes [1, 2, 1].
- In the second pass, everyone in the line buys a ticket and the line becomes [0, 1, 0].
The person at position 2 has successfully bought 2 tickets and it took 3 + 3 = 6 seconds.

Example 2:
Input: tickets = [5,1,1,1], k = 0
Output: 8
Explanation:
- In the first pass, everyone in the line buys a ticket and the line becomes [4, 0, 0, 0].
- In the next 4 passes, only the person in position 0 is buying tickets.
The person at position 0 has successfully bought 5 tickets and it took 4 + 1 + 1 + 1 + 1 = 8 seconds.

9. **Insertion sort**

Sort the array using insertion sort, and then find the **majority element** is the element that appears more than ⌊n / 2⌋ times. You may assume that the majority element always exists in the array.

Example 1:
Input: nums = [3,2,3]
Output: 3
Example 2:
Input: nums = [2,2,1,1,1,2,2]
Output: 2

10. **Merge sort**

Given an array of integers nums, sort the array in ascending order and return it.
Solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

Example 1:
Input: nums = [5,2,3,1]
Output: [1,2,3,5]
Explanation: After sorting the array, the positions of some numbers are not changed (for example, 2 and 3), while the positions of other numbers are changed (for example, 1 and 5).
Example 2:
Input: nums = [5,1,1,2,0,0]
Output: [0,0,1,1,2,5]
Explanation: Note that the values of nums are not necessairly unique.

11. **Hash Table**

Find the first non-repeating character in a string using hash table,Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

Example 1:
Input: s = "leetcode"
Output: 0
Example 2:
Input: s = "loveleetcode"
Output: 2
Example 3:
Input: s = "aabb"
Output: -1

12. Open hashing

Implement the Open Hashing data structure that supports the following operations:

- **insert(key, value)**: Insert a key-value pair into the hash table.
- **get(key)** -> value: Retrieve the value associated with a given key from the hash table.
- **remove(key)**: Remove the key-value pair with the given key from the hash table.

Constraints:

- Use an appropriate hash function to convert keys into valid indices for the hash table.
- If a collision occurs during insertion (i.e., multiple keys map to the same index), use separate chaining to handle it by creating a linked list at that index.

13. Tree traversal algorithms

You are given a binary tree. Your task is to implement three tree traversal algorithms: Preorder Traversal, Inorder Traversal, and Postorder Traversal.

- Preorder Traversal: Visit the root node first, then traverse the left subtree, and finally traverse the right subtree.
- Inorder Traversal: Traverse the left subtree first, then visit the root node, and finally traverse the right subtree.
- Postorder Traversal: Traverse the left subtree first, then traverse the right subtree, and finally visit the root node.

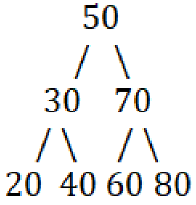
Example :
Input: root = [1,2,3,4,5,null,null]
Output:
Preorder: [1, 2, 4, 5, 3]
Inorder: [4, 2, 5, 1, 3]
Postorder: [4, 5, 2, 3, 1]

14. Binary Search Tree (BST)

The Binary Search Tree must maintain its BST property at all times. For every node in the BST, the keys in its left subtree are less than its key, and the keys in its right subtree are greater than its key. Implement the Binary Search Tree data structure that supports the following operations:

- **insert(key):** Insert a new key into the BST while maintaining its BST property.
- **delete(key):** Delete a key from the BST while preserving its BST property.
- **search(key)** -> bool: Check whether a given key is present in the BST.

Example:
Insertion of elements (50, 30, 20, 40, 70, 60, 80) into the BST.



Constraints:
The BST property is maintained throughout the insertions and deletions.

15. Balanced Binary Tree

Given a binary tree, determine if it is height-balanced. A binary tree is height-balanced if the difference in heights between its left and right subtrees is at most 1 for every node in the tree.Write a function called is_balanced that takes the root of the binary tree as input and returns a boolean value indicating whether the binary tree is height-balanced or not.

Example 1:
Input: root = [3,9,20,null,null,15,7]
Output: true
Example 2:
Input: root = [1,2,2,3,3,null,null,4,4]
Output: false

Example 3:
Input: root = []
Output: true

16. **Graph traversal algorithm**

Implement the Depth First Search (DFS) algorithm to traverse the graph starting from a given source vertex. It that takes the adjacency matrix representation of a graph and the source vertex as inputs and returns a list of vertices visited during the DFS traversal.
Example
adj_matrix = [[0, 1, 0, 1, 0], [1, 0, 1, 0, 1], [0, 1, 0, 1, 0], [1, 0, 1, 0, 1], [0, 1, 0, 1, 0]]
source = 2
DFS traversal order is [2, 1, 0, 3, 4]
Source =0
DFS traversal order is [0, 1, 2, 3, 4]

Indicative list of applications-based experiments

1. Given an expression exp of length n consisting of some brackets, the task is to print the bracket numbers when the expression is being parsed.
   Input:
   The first line contains an integer **T,** the number of test cases. For each test case, there is a string exp containing the expression.
   Output:
   For each test case, the output is the bracket numbers of the expression.
2. Given an unsorted array **arr[]** of size **N**, rotate it by **D** elements(clockwise).
   Input:
   The first line of the input contains **T** denoting the number of test cases. First line of each test case contains two space separated elements, **N** denoting the size of the array and an integer **D** denoting the number size of the rotation. Subsequent line will be the N space separated array elements.
   Output:
   For each test case, in a new line, output the rotated array.
   Exa
   mple
   :
   Inpu
   t:
   2
   5 2
   1 2 3 4 5
   10 3
   2 4 6 8 10 12 14 16 18 20
   Output:
   3 4 5 1 2
   8 10 12 14 16 18 20 2 4 6
3. Given a singly linked list, the task is to find the length of the linked list, where the length is defined as the number of nodes in the linked list.
   Input:
   FirstlineofinputcontainsnumberoftestcasesT.Toreachtestcase,firstlineofinputcontainsnumber of nodes N, to be inserted into the linked list and next line contains data of Nnodes.
   Output:
   There will be a single line of output for each test case, which contains length of the linked list.
4. Given a doubly linked list, rotate the linked list counter-clockwise by P nodes. Here, P is a given positive integer and is smaller than the count of nodes (N) in the linked list.
   Input:
   The first line of input contains an integer **T** denoting the no of test cases. For each test case, the first line of input contains two integers **N and P** denoting the number of nodes in Linked List and the number of nodes to be rotated respectively.
   Output:
   For each test case, output the final linked list after the P rotations in it**.**
5. Given a stack with **push(),pop(),empty()**operations,delete**middle**ofitwithoutusinganyadditional data structure.
   Middle: ceil(size_of_stack/2.0)
   Input Format:

The first line contains an integer **T**, the number of test cases. For each test case, the first line contains an integer **sizeOfStack** denoting the stacksize. Nextline contains space separated integers that will be pushed into the stack.

Output Format:

For each test case, in a new line, print the stack elements. **If stack size if 1 then just print**

6. Given a Queue **Q** containing **N** elements. The task is to reverse the Queue. Your task is to complete the function **rev(),** that reverses the **N** elements of the queue.

   Input Format:

   The first line of input contains an integer **T** denoting the Test cases. Then **T** test cases follow. The first line contains N which is the number of elements which will be reversed. Second line contains N space separated elements.

   Output Format:

   For each test case, in a new line, print the reversed queue.

7. Given a Binary Search Tree, find the sum of all leaf nodes. BST has the following property (duplicate nodesarepossible):The**leftsubtree**ofanodecontainsonlynodeswith**keysless**thanthenode'skey; The **right sub tree** of a node contains only nodes with **keys greater than** or **equal** to the node's key. **Input:**

   The first line of input contains a single integer **T** denoting the number of test cases. Then T test cases follow.Eachtestcaseconsistsoftwolines.Thefirstlineofeachtestcaseconsistsofinteger**N**,denoting the number of elements in the BST. The second line of each test case consists of N space-separated integers denoting the elements in the BST.

   Output:

   For each test case, in a new line, print the sum of leaf nodes.

Reading Material (s)

1. Data Structures Lab manual, Department of CSE, GMRIT, Rajam

21CS308 JAVA Lab

0 0 3 1.5

Course Outcomes

1. Make use of JAVA SDK environment to create - debug and run java programs
2. Create applications based on code reusability
3. Develop programs using threads
4. Develop and debug real time problems using exception handling
5. Create interactive applications using event handling mechanisms
6. Design Graphical User Interface using AWT components

COs	PO2	PO3	PO4	PO5
1	3	2	3	1
2	3	2	3	1
3	2	2	2	1
4	2	2	3	1
5	2	2	3	2
6	2	3	2	2

List of Experiments

1. Case Study: Permutations and Combinations Calculator

User Input:
- The program should prompt the user to enter the total number of elements (n) and the number of elements to be selected (r).
- Validate the input to ensure that both n and r are positive integers, and r is less than or equal to n.

Functional Description:
- Create a method **calculate Permutations** that takes two integers n and r as input and returns the number of permutations.

Formula *nPr = n! / (n - r)!, where "!" denotes factorial.*
- Create a method **calculate Combinations** that takes two integers n and r as input and returns the number of combinations.

Formula *nCr = n! / (r! * (n - r)!).*
- Implement a separate method to calculate the factorial of a given integer.

Sample Output:
```
Enter the total number of elements (n): 5
Enter the number of elements to be selected (r): 3
Permutations (nPr): 60
Combinations (nCr): 10
```

2. Conversion Utility that allows users to convert measurements between different units.

User Input:
- The program should prompt the user to enter positive integer value for conversion type.
- The program should prompt the user to enter the positive input (feet/ meters/ miles/ kilometers) based on user option.

Functional Description:
- utilize method overloading to handle conversions between various units, such as feet to meters, miles to kilometers, and vice versa.

formula:feet to meters - feet * 0.3048
miles to kilometers - miles * 1.60934

Sample Output:
```
Measurement Conversion Utility
1. Feet to Meters
2. Meters to Feet
3. Miles to Kilometers
4. Kilometers to Miles
5. Exit
Enter your choice: 3
Enter the distance in miles: 5
5 miles is approximately 8.046 kilometers.
```

3. Case Study: **BankAccount** Class with Constructor Overloading

Functional Description:

- Create a **BankAccount** class to represent a bank account. The class should have private attributes such as **accountNumber** (a unique identifier for the account), **accountHolderName**, and **balance** (representing the current account balance).
- The class should have the following constructors:
  - **BankAccount(String accountNumber, String accountHolderName)**: Create an account with a zero balance.
  - **BankAccount(String accountNumber, String accountHolderName, double initialBalance)**: Create an account with the specified initial balance.
- Implement methods to:
  - **deposit(double amount):** Add the given amount to the account balance.
  - **withdraw(double amount):** Subtract the given amount from the account balance if sufficient funds are available.

Sample Output:

```
Welcome to the Bank Account System
Creating a new account...
Enter the account holder's name: John Doe
Enter the initial balance: 500.0
Account created successfully.
Account Number: ACCT123456
Account Holder: John Doe
Balance: $500.00
What would you like to do?
1. Deposit
2. Withdraw
3. Exit
Enter your choice: 1
Enter the amount to deposit: 250.0
$250.00 deposited successfully.
Updated Balance: $750.00
```

4. Implement inheritance concept for below case study:

Note: class **FictionBook** extends class **LibraryBook**, class **ClassicBook** and class **ScienceFictionBook** extends from **FictionBook**

- **LibraryBook**: Represents a general library book with attributes like title, author, and year, along with a method **displayBookInfo()** to display book details.
- **FictionBook**: Extends **LibraryBook** and adds an additional attribute genre.
- **ClassicBook**: Extends **FictionBook** and adds an additional attribute publisher.
- **ScienceFictionBook**: Extends **FictionBook** and adds an additional attribute edition

5. Case Study: Abstract Class for Finding Areas of Different Shapes

User Input:

- The program should prompt the user to enter positive integer value to calculate type of area.
- The program should prompt the user to enter the input based on user option.
  - **Circle**: Requires the radius as an input parameter
  - **Rectangle**: Requires the length and width as input parameters.
  - **Triangle**: Requires the base and height as input parameters.
  - **Square**: Requires the side length as an input parameter.

Functional Description:

- Create an abstract class **Shape** that will serve as the base class for different shapes. Include an abstract method **calculateArea()**
- Create concrete subclasses that extend the **Shape** class for different shapes like **Circle**, **Rectangle**, **Triangle** and **Square**.

Sample Output:

```
Select a shape:
1. Circle
2. Rectangle
3. Triangle
4. Square
5. Exit
```

```
Enter your choice: 1

Enter the radius of the circle: 5
The area of the selected shape is: 78.53981633974483
```

6. Implement Dynamic Method Dispatch in Java.
7. i. Create an **ArrayList** of integers to store a collection of numbers. Implement methods to add elements to the list, remove elements, update elements, and search for elements in the list.

   ii. Create a Java program that accepts a string input from the user. Use the **StringTokenizer** class to tokenize the input string into separate tokens based on specified delimiters (e.g., space, comma, etc.).
8. Demonstrating Mutability Differences between **String**, **StringBuilder**, and **StringBuffer** in Java
   - Explain the Concept in 100 words.
   - Perform string manipulation operation concatenation on each class
   - Print the results of each class after modification operation
9. Demonstrate multiple inheritance-like behavior achieved through interface implementation.
10. Case Study: Java Program Illustrating Multiple Catch Clauses with Finally Block

    Functional Description:
    - Implement exception handling for possible exceptions that may occur during the division, such as **ArithmeticException** and **NumberFormatException**.
    - Use multiple catch clauses to handle **ArithmeticException** and **NumberFormatException** separately.
    - Use a **finally** block to execute code that should be performed regardless of whether an exception occurred or not

    Sample Output:

    ```
    Enter the dividend: 10
    Enter the divisor: 0
    Division by zero is not allowed.
    Execution completed.
    ```
11. Create your own custom exception classes to handle specific situations or errors that are not covered by the built-in exception classes.

    Functional Description:
    - Create a custom exception class that extends the **Exception** class (or any other appropriate superclass, depending on your requirements).
    - Define the custom exception class to represent a specific error situation, such as **InvalidAgeException**, which is thrown when an invalid age is provided.
    - Use the custom exception class in a method to throw the exception when a certain condition is met, such as when the age is less than zero or exceeds a specific limit.

    Constraints:

    The program should prompt the user to enter age integer value. Here, age<0 or age>120

    Sample Output:

    ```
    Enter age:-5
    Exception: Invalid age. Age must be between 0 and 120.
    ```
12. Creates three threads by extending the **Thread** class. Each thread will display a specific message at a regular interval. The messages to be displayed are as follows:

    First thread: "**Good Morning**" every 1 second

    Second thread: "**Hello**" every 2 seconds

    Third thread: "**Welcome**" every 3 seconds

    Note: implement the same by using **Runnable** interface.
13. Case Study: Java Program for Producer-Consumer Problem

    User Input:
    - The program should prompt the user to enter positive integer valuesfor **bufferSize**
    - The program should prompt the user to enter positive integer values to **produceCount** and **consumeCount**

    Functional Description:
    - Buffer Implementation:
      - Create a shared buffer with a limited size (e.g., an array or an ArrayList) to hold the produced items.
    - Producer & Consumer Implementation:
      - Create a Producer class & Consumer class that implements the Runnable interface.
      - The Producer class should produce items and put them into the buffer.
      - The Consumer class should consume items from the buffer.
      - Implement synchronization to avoid race conditions when accessing the buffer.

Sample Output:

```
Enter Buffer Size:5

Enter Procedure Count:5
Enter Consumer Count:5
Produced: 0
Consumed: 0
Produced: 1
Consumed: 1
Produced: 2
Consumed: 2
Produced: 3
Consumed: 3
Produced: 4
Consumed: 4
```

14. Case Study: Java Program for a Basic Calculator using AWT

In this case study, we'll implement a Java program to create a basic calculator using AWT. The calculator will have a simple graphical user interface (GUI) with buttons for numbers, arithmetic operations, and a display to show the result of calculations.

AWT Implementation:

- Create a Java AWT to implement the calculator GUI.
- The AWT should have buttons for digits (0 to 9), arithmetic operations (addition, subtraction, multiplication, division), and an equal (=) button to perform calculations.

Display and Input:

- Implement a display area where the user can see the entered numbers and the result of calculations.
- Provide a way for users to input numbers and arithmetic operations using the GUI buttons.

Basic Calculations:

- Implement methods to handle basic arithmetic calculations (addition, subtraction, multiplication, division).
- Perform calculations based on the user's input and display the result in the GUI.

User Interface:

- Design a user-friendly GUI with appropriate labels, buttons, and display areas.
- Use layout managers to organize the components effectively.

15. Case Study: Java Program for Temperature Converter using AWT

In this case study, we'll implement a Java AWT to create a Temperature Converter. The AWT will allow users to convert temperatures between Celsius and Fahrenheit. The user will input a temperature value in one unit, and the AWT will display the converted temperature in the other unit.

AWT Implementation:

- Create a Java AWTpplet to implement the temperature converter GUI.
- The AWT should have input fields for the temperature value and buttons to select the source and target temperature units (Celsius or Fahrenheit).
- Use appropriate layout managers to organize the components effectively.

Conversion Logic:

- Implement methods to handle temperature conversions between Celsius and Fahrenheit.
- The temperature conversion formulas are:
  - Fahrenheit to Celsius: C = (F - 32) * 5 / 9
  - Celsius to Fahrenheit: F = (C * 9 / 5) + 32

User Interface:

- Design a user-friendly GUI with appropriate labels, input fields, and buttons.
- Display the converted temperature in the GUI after the conversion is performed.

16. Case Study: Java Swing Application for Job Application Form

In this case study, we'll implement a Java Swing application with a JFrame that contains fields for name, age, email, and qualifications. The application will validate the form data when the user clicks the "Submit" button. If all fields are filled, the data will be written to a file named "application_form.txt."

Swing Application:

- Create a Java Swing application with a JFrame to display the job application form.
- Design the form with labels for each field (name, age, email, qualifications) and corresponding input fields (text fields or other appropriate components).
- Add a "Submit" button to the form to allow users to submit their job application.

Data Validation:

- Implement validation for the form data to ensure that all fields are filled before submission.

- Display error messages for missing fields or invalid data.

Data Writing:

- Create a method to write the validated form data to a file named "application_form.txt."
- The file should be created in the current working directory.
- Use appropriate file handling techniques to write the data.

User Interaction:

- Allow users to input their name, age, email, and qualifications using the form fields.
- Trigger the form data validation when the user clicks the "Submit" button.
- Display appropriate messages to inform the user about the success or failure of form submission.

List of Augmented Experiments

1. New Patient Registry Management System
2. Restaurant Billing Management System
3. Library Management System
4. ATM Management System
5. Bus Ticket Booking Management System
6. Movie Ticket Booking Management System
7. Queuing Management System
8. Attendance Management System
9. Medical Store Billing Management System
10. Text Editor Projects in Java
11. Google Search Engine Filter
12. Electronic voting System
13. Day Planner
14. Library management System
15. Personal Finance Management System

Reading Material (s)

1. JAVA Lab manual, Department of CSE and IT, GMRIT, Rajam

21ESX01 Employability Skills I

0 0 2 0

Course Outcomes

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Introspect & develop life skills with constructive approach
3. Assess and improve analytical skills
4. Illustrate the knowledge of the basics of Unix/Linux commands
5. Apply the concepts of shell scripting on data and files
6. Understand the utility of SED to work on specific data

COs –POs Mapping

COs	PO1	PO2	PO5	PO 8	PO10	PO12
CO1	-	-	-	-	3	-
CO2	-	-	-	-	2	2
CO3	2	-	-	2	-	-
CO4	3	-	2	2	-	-
CO5	3	2	2	-	-	2
CO6	2	2	2	-	-	-

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Unit-I

Communication Skills **16 hours**

Communication Skills & Confidence: How Communication Skills affect Confidence? How to communicate effectively.(with Examples)

Listening: Listening? , Listening Vs Hearing, Possible reasons for why people do not Listen at times, Active Listening Vs Passive Listening, How Listening can affect our relationships? How Listening helps in Campus Placements also? (with Examples)

Goal Setting: SMART Technique to Goal Setting, Putting First things First, SWOT Analysis and Time Management

Attitude & Gratitude: Attitude Vs Skills Vs Knowledge, Attitude Vs Behaviour, How to develop Positive Attitude? Developing the attitude of Gratitude.

Public Speaking: JAM, J2M, Presentations by Students on General Topics.

Quantitative Aptitude **16 hours**

Number system, L.C.M and H.C.F, Problems on Ages, Averages, Time and work, Pipes and cisterns

Unit-II

Module-1:

No of Hours: 16 Hours

Introduction: Importance of Competitive Programming, Conditional Statements, Loops, Nested Loops. Strings: Declaration of strings, Accessing strings elements, operations

Practice Problems:

Week	Topic	Practice Problem Links
1	Introduction to Competitive Programming	**Codechef, Hackerrank,Codeforces, Leetcode, interviewbit…etc**
2	Conditional Statements-I	https://www.codechef.com/practice/tags/basic-programming-concepts
		https://www.codechef.com/practice/tags/conditional-statements
3	Conditional Statements-II	https://www.codechef.com/practice/tags/basic-programming-concepts
		https://www.codechef.com/practice/tags/conditional-statements
4	Loops-I	https://www.codechef.com/practice/tags/loops
		https://leetcode.com/tag/math/
		https://www.interviewbit.com/practice/#topics[]=math
		https://www.codechef.com/practice/tags/basic-maths
5	Loops-II	https://www.codechef.com/practice/tags/loops
		https://leetcode.com/tag/math/
		https://www.interviewbit.com/practice/#topics[]=math
		https://www.codechef.com/practice/tags/basic-maths
6	Nested Loops-I	https://www.codechef.com/practice/topics/math
		https://www.interviewbit.com/practice/#topics[]=math
		https://leetcode.com/tag/math/

7	Nested Loops-II	https://www.codechef.com/practice/topics/math https://www.interviewbit.com/practice/#topics[]=math https://leetcode.com/tag/math/
8	Strings-I	https://www.codechef.com/practice/topics/strings https://leetcode.com/tag/string/ https://www.interviewbit.com/practice/#topics[]=strings
9	Strings-II	https://www.codechef.com/practice/topics/strings https://leetcode.com/tag/string/ https://www.interviewbit.com/practice/#topics[]=strings
10	String Matching Algorithms	https://www.codechef.com/practice/topics/strings https://leetcode.com/tag/string/ https://leetcode.com/tag/string-matching/ https://www.interviewbit.com/practice/#topics[]=strings

Module-2:

No of Hours: 16 Hours

Arrays: Basics on Arrays: Creating elements, converting a string into array of characters, converting a number into array of character, accessing array elements **Recursion:** Definition, examples and how recursion differs from iteration.

Practice Problems:

Week	Topic	Practice Problem Links
11	Arrays - I	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=arrays https://leetcode.com/tag/array/ https://www.codechef.com/practice/topics/arrays https://www.interviewbit.com/practice/#topics[]=arrays
12	Arrays - II	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=arrays https://leetcode.com/tag/array/ https://www.codechef.com/practice/topics/arrays https://www.interviewbit.com/practice/#topics[]=arrays
13	2D Arrays & Matrix	https://www.interviewbit.com/practice/#topics[]=two-pointers https://leetcode.com/tag/two-pointers/ https://leetcode.com/tag/matrix/ https://www.codechef.com/practice/tags/matrices
14	Recursion - I	https://leetcode.com/tag/recursion/ https://www.codechef.com/practice/tags/recursion
15	Recursion - II & Assessment	https://leetcode.com/tag/recursion/ https://www.codechef.com/practice/tags/recursion https://www.hackerearth.com/practice/basic-programming/recursion/recursion-and-backtracking/practice-problems/
16	Recursion - III & Assessment - II	https://leetcode.com/tag/recursion/ https://www.codechef.com/practice/tags/recursion

Total: 32 Hours

References:

1. https://leetcode.com
2. https://www.codechef.com
3. https://www.hackerearth.com
4. https://www.interviewbit.com
5. https://www.hackerrank.com

21HSX11 CC & EC Activities I

0 0 1 0

Course Outcomes

1. Interpret and present the abstractive technical information through an activity
2. Think critically in providing solutions to the generic and common problems
3. Demonstrate the creative thinking in dealing with liberal arts
4. Instill team sprit through active engagement with the peer
5. Develop programs of common interest having social impact
6. Empower the under privileged through motivational activities

COs –POs Mapping

COs	PO6	PO7	PO9	PO10
1				3
2	3	2		
3	3			
4			3	
5	3			
6	3			

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Co-Curricular and Extra Curricular (CCEC) Activities:

Students shall acquire 1 credit each in 2nd and 3rd years with the following scheme:
Scheme of evaluation for the CCEC activities:
• No. of slots in each Semester @ 1 slots every week : 12
• No. of Streams (1 CC + 1 EC + 1 Community Engagement) : 3
• No. of slots allotted for each stream : 4
Requirement for the award of 1-Credit
• Students shall choose at least two streams of events in each semester and among which Community engagement through NSS is mandatory
• Students shall secure an overall attendance of 75% in all events put together

The credits earned through these courses will be indicated in the grade memo and will be taken into account for CGPA calculation.

4thSemester

21IT304 Database Management Systems

3 0 0 3

Course Outcomes

1. Understand the fundamental concepts of data base and data models
2. Explain the use of Relational Algebra and integrity constraints in databases
3. Use SQL's Commands to handle the Database
4. Apply Normalization for schema refinement
5. Make use of the concept of transaction management and recovery system in databases
6. Outline Indexing concepts, different types of data

CO–PO Mapping

CO	PO1	PO4	PO12
1	3	2	1
2	3	3	2
3	3	2	2
4	2	3	1
5	3	3	2
6	3	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I

Introduction to DBMS and ER Model **12 Hours**

DBMS Vs. File System, instance and schema, Data abstraction, Data independence, database users and database administrator, Database system structure, Introduction to Data Models (E-R Model, Relational Model, Hierarchical Model, Network Model, Object Oriented Data Model), Database Design Process, Entities, Attributes, Entity Sets, Relationships, Relationship Sets, Additional features of ER Model.

Applications of DBMS, Object Relational Data Model

Unit II

Introduction to Relational Model and Basic SQL Queries **12 Hours**

Relational Algebra Operations: Selection, Projection, Rename, Set Operators, Joins, Division, Examples of Relational Algebra Queries, Relational Calculus: Tuple Relational Calculus.

Integrity Constraints over Relations, Introduction to Views.

SQL Queries: Basic Structure, Set Operations, Aggregate Functions, Null values, Sub Queries, Group By And Having Clauses, Outer Joins.

Domain Relational Calculus, Query Optimization

Unit III **12 Hours**

Normalization and Transaction Management

Introduction To Schema Refinement - Problems Caused By Redundancy - Decomposition - Problems Related To Decomposition - Functional Dependency - Closure of a Set of Fds - Attribute Closure - First - Second - Third Normal Forms – BCNF - Multi Valued Dependencies – Fourth Normal Form, Join Dependency, Fifth Normal Form

Transactions: Acid Properties of Transaction - Transaction States - Schedule: Serial Schedule - Concurrent Schedules - Anomalies Associated with Concurrent Schedules (RW - WR - and WW Conflicts) - Serializability - Conflict Serializability - and View Serializability.

EF Codd Rules, Domain Dependency

Unit IV **12 Hours**

Locking, Recovery Systems, Indexing, Different Types of Data

Introduction to Lock Management-Lock Based Concurrency Control: 2pl-Strict 2pl-Concurrency without Locking: Timestamp–Based Concurrency Control, Optimistic Concurrency Control. Introduction to Aries - the Log - the Write-Ahead Log Protocol-Check Pointing Indexing: Types of Single-Level Ordered Indexes, Multilevel Indexes Different Types of Data: Structured, Semi-Structured and Unstructured Data

Heap File, Hash File Organizations

Total: 48 Hours

Textbook (s)

1. Elmasri & Navatha, Fundamentals of Database Systems, Pearson Education, 7thEdition,2016
2. SilberschatzKorth, Database System Concepts, McGraw hill, 7thEdition,2020

Reference (s)

1. SorayaSedkaoui, Data Analytics and Big Data, Wiley, 1st Edition,2018.
2. PeterRob&CarlosCoronel,DatabaseSystemsdesign,ImplementationandManagement,9th Edition, 2010.
3. Raghurama Krishnan & Johannes Gehrke, Database Management Systems, TATA McGraw-Hill, 3rd Edition,2003
4. C.J.Date, An Introduction to Database Systems, Pearson Education, 8th Edition,2006

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	40	40	--
Understand	30	40	--
Apply	30	20	50
Analyze	--	--	50
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. List any four application of DBMS
2. Define data model
3. List any four applications for triggers
4. Define functional dependency
5. List the 4 properties of Transaction

Understand

1. Explain E-R Model with suitable example
2. Explain the role of integrity constraints in database design
3. Illustrate the working principle of 'write a head log' protocol
4. Differentiate 3NF and4NF
5. Explain Two Phase Locking Protocol

Apply

1. When multiple transactions are being executed by the operating system in a multiprogramming environment, there are possibilities that instructions of one transaction are interleaved with some other transaction. Apply the suitable concept to overcome the problem
2. Classify various normal forms according to their applicability
3. Give some real-world applications of Normalization
4. Illustrate the Commit and Rollback operations of Transaction Control
5. Give some real-world applications for Database indexing techniques

Analyze

1. Compare File processing system with DBMS
2. Analyze different locking protocol for concurrency control and serializability
3. Normalization will increase the complexity of the database design. Justify
4. Compare DDL and DML of SQL
5. Compare and Contrast Serializability and Recoverability

Evaluate

1. Is database redesign is necessary? explain
2. How can you evaluate the performance of two data models?
3. Evaluate the performance of query processor and list the corresponding metrics
4. How can you assess the throughput and delay for any DBMS?

5. How can you evaluate the impact of data models on the query processing?

Open Book Exam Questions

Question 1:

Anitha has a large CD collection. Her friends like to borrow her CD's, and she has to keep track of who has what. She maintains a list of friends, identified by unique FID's and a list of CD's, identified by CID's. With each friend are the name and telephone numbers which she can call to get the CD back. With each CD is actor name and title. Whenever a friend borrows a CD, She will enter that fact into her database along with the date borrowed. Whenever the CD gets returned, that fact, too, gets noted along with the date returned. Anitha wants to keep a complete history of her friends' borrowing habits so that she can ask favors of the heavy borrowers.

Draw an ER diagram to figure out the above situation and identify types of attributes and cardinality. Represent this database as a collection of 3NF relational tables.

Question 2:

The relational scheme R(A,B,C,D,E,F) and set of functional dependencies AB -> D, E -> C, AF -> B. From this, find out all super keys for this relation, and which of these super keys form a key.

21IT403 Operating Systems

Course Outcomes **3 0 0 3**

1. Understand computer resources and operating system management.
2. Analyze various CPU Scheduling Algorithms for Process Management.
3. Examine process synchronization and coordination of operating system.
4. Analyze the Main Memory Management and allocation strategies.
5. Identify the use of Virtual Memory management policies with respect to storage management.
6. Identify the need of File-System Interface and I/O Systems.

CO–PO Mapping

CO	PO1	PO12
1	3	2
2	3	2
3	3	2
4	3	2
5	3	2
6	3	1

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I
Operating-Systems Overview and Process Management **12 Hours**

Operating-System Overview: Computer-System Organization and Architecture, Operating-System Structure, Operating-System Operations & Services, System Calls & its types.

Threads: Multi Core Programming, Multithreading Models, Thread Scheduling algorithms.

Process Management: Process Concepts, Process Scheduling Criteria, Scheduling Algorithms and evaluation.

Thread issues, Multilevel Queue, Multilevel feedback Queue Scheduling.

Unit II
Inter Process Communication Mechanism **12 Hours**

Process Synchronization: Cooperative process, the Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Synchronization problems.

Deadlocks: Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance & Detection, Recovery from Deadlock.

Monitors, Synchronization Examples.

Unit III
Memory Management **12 Hours**

Main Memory: Contiguous Memory allocation, Swapping, Segmentation, Paging, Segmented paging, Multilevel paging.

Virtual Memory Management: Demand Paging, Page Replacement algorithms, Allocation of Frames.

Structure of page table, Thrashing, Memory-Mapped Files.

Unit IV
File System Interface &I/O Systems **12 Hours**

Mass-Storage Structure: Disk structure, Disk Scheduling, Disk management, Raid Structure.

File System: Access Methods, Directory Structures, Allocation Methods, Free-Space Management.

I/O Systems: I/O hardware Application of I/O Interface, Kernel I/O Sub-System.

File Sharing, File System Recovery, Transforming I/O Requests to Hardware Operation.

Total: 48 Hours

Textbook (s)

1. Operating System Concepts, Abraham Silberschatz, Greg Gagne, Peter B. Galvin, 9th Edition, Wiley, 2016.
2. Operating Systems, Harvey M. Deitel,Paul J. Deitel, David R. Choffnes, 3rd Edition, Pearson Prentice Hall,2004.

Reference (s)

1. Operating Systems: Internals and Design Principles, William Stallings, 7th Edition, Pearson Prentice Hall,2013.
2. Operating systems: A Concept based Approach, D. M. Dhamdhere, 2nd Edition, TMH,2006.
3. Operating System: A Design Approach, Crowley, 1st Edition, TMH,2001.
4. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI,2009.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	20	20	--
Understand	30	10	--
Apply	30	30	40
Analyze	10	20	30
Evaluate	10	20	30
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. List any four operating systems
2. Define operating system
3. List any four operating system services

Understand

1. Explain Systemcalls
2. Explain the role memory management in operating system
3. Illustrate the working principle critical section problem

Apply

1. When multiple transactions are being executed by the operating system in a multiprogramming environment, there are possibilities that instructions of one transaction are interleaved with some other transaction. Apply the suitable concept to overcome the problem
2. Give an example of a scenario that might benefit from a file system supporting an append-only access write.

Analyze

1. Context switching between two threads of execution within the operating system is usually performed by a small assembly language function. In general terms, what does this small function do internally?
2. Compare CPU scheduling algorithms
3. Analyze the general strategy behind deadlock prevention, and give an example of a practical deadlock prevention method.

Evaluate

1. Assuming the operating system detects the system is deadlocked, what can the operating system do to recover from deadlock?
2. Describe how to implement a lock using semaphores

Open Book Exam Questions

1. A file to be shared among different processes, each of which has a unique number. The file can be accessed simultaneously by several processes, subject to the following constraint: the sum of all unique numbers associated with all processes currently accessing the file must be less than n. Write a monitor to co-ordinate the access to the file. On may want to write start_access and end_access monitor procedures.

2. Consider a paging system with the page table stored in memory:
   a. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
   b. If we add associative registers, and 75% of all page-table references are found in the associative registers, what is the effective memory reference time?

21CS403 Computer Organization and Architecture

3 0 0 3

Course Outcomes

At the end of the course, students will be able to

1. Interpret the functional architecture of computing systems.(Understand).
2. Summarizethetypesofinstructionanditsmicrooperationwithaddressingmodes(Understand)
3. Identify various arithmetic operations on fixed, floating point numbers and its representation (Apply)
4. Illustrate the concepts of control unit design and I/O processor(Understand)
5. Understand the memory hierarchy concepts(Understand)
6. Describe concept of parallelism and types of hazard(Understand)

CO-PO Mapping

CO	PO1	PO12
1	3	2
2	2	2
3	3	1
4	3	2
5	3	2
6	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I

Overview & Microoperation **13 Hours**

Components of a computer system – Performance measures - Classifying Instruction Set Architecture-Representing instructions -Micro operation – Logical operations – Shift operations - instruction codes - Computer Registers instruction –memory Reference instruction –Input-Output Reference instruction - Instruction cycle -Addressing and addressing modes.

Trends in Technology-Arithmetic micro-operations

Unit II

Arithmetic Operations **11 Hours**

ALU - Addition and subtraction with Signed Magnitude Data - Hardware Implementation – Multiplication – Hardware Implementation for Signed Magnitude Data – Division - Hardware Implementation for Signed Magnitude Data – Divide Overflow - Floating Point operations – Parallelism and Computer Arithmetic: Sub wordParallelism.

BCD Adder-BCD Subtraction

UNIT III

Control Unit and Memory Systems **11 Hours**

Basic MIPS implementation – Building data path – Control Implementation scheme – Memory hierarchy – Cache basics – Measuring and improving cache performance - Virtual memory- Input/output system-programmed I/O-DMA and Interrupts-I/O processors

Stack organization-RISC Vs CISC Architecture

Unit IV

Parallelism **13 Hours**

Instruction-level-parallelism – Parallel processing challenges – Flynn's classification – Multicore processors- Pipelining – Arithmetic pipeline –Instruction pipeline -Pipelined data path and control – Handling Data hazards & Control hazards – Exceptions.

Vector processing –single processor Vs parallel processor

Total: 48 Hours

Text Book(s):

1. DavidA.PattersonandJohnL.Hennessey,"Computerorganizationanddesign:Thehardware /software interface‟, Morgan Kauffman / Elsevier, Fifth edition, 2014.
2. M.Morris Mano," Computer System Architecture", 3rdedition, Pearson/PHI,2092.

Reference(s):

1. V.CarlHamacher, Zvonko G. Varanesic and Safat G. Zaky, "Computer Organization ", 6thedition, McGraw-Hill Inc,2012.
2. William Stallings "Computer Organization and Architecture, Seventh Edition, Pearson Education, 2007.
3. Andrew S Tanenbaum "Structured Computer Organization ", 5thedition, Pearson/PHI,2007

Internal Assessment Pattern

Cognitive Level	**Int. Test 1 (%)**	**Int. Test 2 (%)**	**Open book Test[1] (%)**
Remember	40	--	--
Understand	40	50	--
Apply	20	50	80
Analyze	--	--	20
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. What is micro operation?
2. Show the Register Reference Instruction format.
3. Define PC and MAR.
4. What are the two types of data representation?
5. Define Associative Memory

Understand

1. Identify the basic functional units of the system
2. Explain about logic micro operations and its applications with examples
3. Differentiate RISC and CISC architecture in terms of their instruction set and addressing modes.
4. Compare hardwired control unit is differing from micro programmed control unit designs
5. Demonstrate control memory

Apply

1. Starting from an initial value of R=11011101, determine the sequence of binary values in R after a logical shift-left, followed by a logical shift-right and a circular shift-right.
2. Analyze the contents of Register A that holds 8 bit binary 11011001 and Determine the B-operand and the logic micro operation to be performed in order to change the value in A to: (i) 01101101 (ii) 11111101State the differences between register stack and memory stack.
3. Perform the arithmetic operations (+70) + (+80) and (-70) + (-80) with binary numbers in signed-2's complement representation. Use eight bits to accommodate each number together with its sign. Show that overflow occurs in both cases, that the last two carries are unequal, and that there is a sign reversal.
4. Show the hardware to be used for the addition and subtraction of two decimal numbers with negative numbers in signed- 10's complement representation. Indicate how an overflow is detected. Derive the flowchart algorithm and try a few numbers to convince yourself that the algorithm produces correct results.

5. The procedure for aligning mantissas during addition or subtraction of floating-point numbers can be stated as follows: Subtract the smaller exponent from the larger and shift right the mantissa having the smaller exponent a number of places equal to the difference between the exponents. The exponent of the sum (or difference) is equal to the larger exponents. Without using a magnitude comparator, assuming biased exponents, and taking into account that only the AC can be shifted, derive an algorithm in flowchart form for aligning the mantissas and placing the larger exponent in the AC

Open book questions

Apply

1. SmithandGoodmanfoundthatforagivensmallsize,adirect–mappedinstructioncacheconsistently outperformed a fully associative instruction cache using LRU replacement.

a. Explain how this would be possible (*Hint:* You can't explain this with the three C's model because it "ignores" replacement policy)
b. Explain where replacement policy fits into the three C's model, and explain why this means that misses caused by a replacement policy are "ignored"- or, more precisely, cannot in general be definitively classified by the three C'smodel.
c. Are there any replacement polices for the fully associative cache that would outperform the direct-mapped cache? Ignore the policy of "do what a direct- mapped cache would do".
d. Use a cache simulator to see if Smith and Goodman's results hold for memory reference traces that you have access to. If they do not hold, why not?

2. John takes two numbers in sign magnitude representation (the two numbers are same with different signs), The 1's complement of one number is 6. The difference between 1's complement of these two numbers is 32. Find the numbers. And also find the product of these two numbers using the result of 2's complement value of these two numbers.

Analyze

1.A two-word instruction is stored in memory at an address designated by the symbol W. The address field of the instruction (stored at W + 1) is designated by the symbol Y. The operand used during the executionoftheinstructionisstoredatanaddresssymbolizedbyZ.Anindexregistercontainsthevalue X. State how Z is calculated from the other addresses if the addressing mode of the instruction is a.direct a. indirect c. relative d.indexed.

2. An 8-bit computer has a 16-bit address bus. The first 15 lines of the address are used to select a bank of 32K bytes of memory. The high-order bit of the address is used to select a register which receives the contents of the data bus. Explain how this configuration can be used to extend the memory capacity of the system to eight banks of 32K bytes each, for a total of 256K bytes of memory.

21CS404 Design and Analysis of Algorithms

3 0 2 4

Course Outcomes

1. Understand the fundamentals for analyzing time and space complexity of algorithms
2. Describe the basic principles of various algorithm design techniques
3. Choose and apply appropriate algorithm design technique to solve real time application
4. Analyze the performance of algorithm designed for real time application
5. Understand P and NP problems
6. Analyze the performance of algorithm designed for real time application

COs–POs Mapping

CO	PO2	PO3
1	3	2
2	2	2
3	3	3
4	3	3
5	2	2
6	3	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12 +8 hours**

Introduction to Algorithms

Fundamentals of algorithmic problem solving – Analysis framework - Performance Analysis: - Space complexity, Time complexity - Growth of Functions: Asymptotic Notation- Big oh notation, Omega notation, Theta notation, little oh, little omega. Mathematical Analysis of Non-recursive algorithms – Mathematical Analysis of Recursive algorithms - Brute Force: Bubble sort – Sequential search - String matching

Time complexities of basic operations on array, stack, queue

Practical Components

1. Analysis of recursive algorithms
2. Analysis of Non-recursive algorithms
3. Solve the String matching problem using Brute Force approach
4. Solve the Sorting problem using Brute Force approach

Unit II **12+8 hours**

Algorithm Design Techniques - I

Decrease and Conquer: General method, Insertion sort, Topological sorting – Divide and Conquer: General method, Merge sort, Quick sort, Fibonacci search – Transform and conquer: General method, Heap sort – Disjoint sets: Operations, Simple Union and simple find, Weighted Union, CollapseFind.

Graph representation, graph applications

Practical Components

1. Find the Topological order for the given graph using Decrease and Conquer approach
2. Solve sorting problem using divide and conquer approach
3. Solve sorting problem transform and conquer approach
4. Implement Simple Union and Find operation on disjoint-sets

Unit III **12+10 hours**

Algorithm Design Techniques - II

Greedy method: General method, applications:-Job sequencing with deadlines, Minimum cost spanning trees (Prim's and Kruskal's Algorithms), Single source shortest path problem (Dijkstra's Algorithm), Optimal Merge Patterns (two way merge pattern) - Dynamic Programming: General method, applications:-Matrix chain multiplication, 0/1 knapsack problem, all pairs shortest path problem.

Optimal BST, Huffman trees and codes

Practical Components

1. Find the minimum cost spanning tree for a given graph using greedy approach
2. Implement dijkstra's algorithm using greedy approach
3. Find the minimum record movements for optimal merge pattern using greedy approach
4. Find the optimal parenthesization solution for matrix chain multiplication problem using dynamic programming
5. Solve the 0/1 knapsack using dynamic programming

Unit IV **12+ 6 hours**

Limitations of algorithm power methods to cope up

P, NP and NP-Complete Problems: - P and NP problems, NP-Complete problems. Backtracking: General method, applications:-n-queen problem, sum of subsets problem, Hamiltonian Circuit problem. Branch and Bound: General method, Applications: - Traveling sales person problem, Assignment Problem
0/1 knapsack problem, graph coloring problem

Practical Components

1. Find the sum of subsets using backtracking technique
2. Find the solution for Traveling sales person problem using Branch and Bound method
3. Find the solution for Assignment Problem using branch and bound method

Total: 48+32 hours

Textbook (s)

1. Introduction to The Design and Analysis of Algorithms, 3rdEdition, AnanyLevitin, Pearson Education,2017.
2. Introduction to Algorithms, second edition, T.H.Cormen, C.E.Leiserson, R.L. Rivest, and C.Stein, PHI Pvt. Ltd./ Pearson Education
3. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahni and Rajasekaran, University press.

Reference (s)

1. Design and Analysis of algorithms, Aho,Ullman and Hopcroft, Pearsoneducation.
2. Algorithms–Richard Johnson Baughand Marcus Schaefer,Pearson Education.
3. www.geeksforgeeks.org
4. www.hackerearth.com
5. www.tutorialspoint.com

SAMPLE QUESTION (S)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Assignment Test^2 (%)
Remember	15	10	-
Understand	35	20	10
Apply	30	40	60
Analyze	20	30	30
Evaluate	--	-	-
Create	--	-	-
Total (%)	100	100	100

Remember

1. Define performance analysis of an algorithm.
2. Define recurrence relation
3. Define disjoint sets.
4. Define optimality principle

Understand

1. Explain back tracking algorithm
2. Explain Asymptotic notations
3. Explain the LC search algorithm
4. Explain the divide and conquer strategy. Write the applications of divide and conquer strategy and write the control abstraction for divide and conquer strategy
5. Explain the 4 cases of master theorem

6. What are implicit and explicit constraints in back tracking?

Apply

1. Solve the fractional knapsack problem with capacity m=20,W=(18,15,10),P=(25,24,15) for the optimum solution.
2. Create two arrays that store all even no’s in one array and all odd no’s in other array for a given set of elements
3. Find the space and time complexity of an algorithm to find the maximum element from an array.
4. Solve the recurrence relation T(n)=2T(n/2)+nlogn) using master theorem.
5. Demonstrate the process of Fibonacci search with an example
6. Find the shortest tour for the TSP for the given graph using FIFOBB.

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

Analyze

1. Analyze the time complexity of quicksort.
2. Differentiate greedy method with dynamic approach
3. Discuss the worst case time complexity of travelling sales person problem
4. How polynomial multiplication is best with FFT
5. Compute the time complexity

```
offun()? int fun(int n)
{ int count = 0;
for (int i = n; i > 0; i /=
   2) for (int j = 0; j < i;
   j++)
       count +=
 1; return
 count;
}
```

21CS405 Foundations of Data Science

3 0 1 3

Course Outcomes

1. Understand the pipeline of machine learning to apply on real world problems
2. Understand the need of Similarity and Dissimilarity between features in data
3. Importance of Pre-processing techniques in real-time
4. Identifying the characteristics of various Regression models in real-world
5. Identifying the characteristics of various supervised learning and ensemble models
6. Identifying the characteristics of various algorithms of unsupervised learning

COs–POs Mapping

CO	PO1	PO2	PSO1	PSO2
1	3	2	1	1
2	3	2	2	2
3	2	2	2	2
4	2	3	3	2
5	2	3	3	2
6	3	3	3	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

UNIT I **12 hours**

Introduction: Relation among AI, ML and Data Science, Importance of Data Science, Categorization of Data Science Algorithms, Usecases; **Data Science Process; Data Exploration:** Objectives of Data Exploration, Forms of Data (Structured, Semi Structured, Unstructured), Datasets (data objects and types of attributes/fields), Characteristics of Datasets and corresponding Statistical Measures; **Data Visualization:** Univariate Visualization, Multivariate Visualization, Visualization of Complex Data and Relations(High-Dimensional Data)

Discriminant Analysis, LDA

UNIT II **12 hours**

Data Similarity/Dissimilarity: Data Matrix, Dissimilarity Matrix, Proximity/Similarity Measures for various types of data (nominal, ordinal, binary, numerical, etc.), Dissimilarity for Attributes of Mixed Types, Cosine Similarity; **Data Preprocessing:** Preprocessing Pipeline, Forms of Preprocessing, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Discretization.

Distance based similarity (Euclidean distance, Jaccard Similarity)

UNIT III **12 hours**

Regression (forecast a given numerical quantity): Linear Regression, Logistic Regression (Regression for Classification), Regularization; **Classification:** Classification Principles, Classification Model Evaluation Metrics (Confusion Matrix, ROC and AUC, Lift Covers), Classification using Decision Trees, Bayesian Classification, Distance based Classifier(k-NN), Support Vector Machines (SVM), ANN-based Classification;

LASSO and Ridge Regression

UNIT IV **12 hours**

Ensemble Learning: Conditions for Ensemble Modeling, Ensemble Techniques (Voting, Bagging, Boosting and Random Forest); **Clustering:** Clustering Principles, Applications/Purpose of Clustering- Clustering for description/preprocessing/classification, Types of Clustering, Clustering Evaluation Parameters, Clustering Algorithms (k-Means, Agglomerative/Divisive, DBSCAN and Self Organizing Maps) and Evaluation Metrics; **Data Science Tools: Examples;**

ADABoost, XGBoost **Total: 48 hours**

Textbooks:

1. Vijay Kotu&Bala Deshpande, “Data Science: Concepts and Practice”, 2nd Edition, Morgan Kaufmann Publication, 2017.
2. JIawei Han & Micheline Kamber, “Data Mining: Concepts and Techniques”, 3rd Edition, Morgan Kaufmann Publishers, 2012
3. Pang-Ning Tan et al., “Introduction to Data Mining”, 2nd Edition, Pearson Publications, 2019.

References:

1. Vijay Kotu&Bala Deshpande, “Data Science: Concepts and Practice”, 2nd Edition, Morgan Kaufmann Publication, 2017.
2. Tom M. Mitchell, "Machine Learning ",Tata McGraw Hill, 1997
3. SunilaGollapudi, “Practical Machine Learning” First Edition 2016,Packt Publishing Ltd.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open book Test[1] (%)
Remember	20	20	--
Understand	50	30	--
Apply	30	30	80
Analyze	--	20	20
Evaluate	--	--	--
Create	--	--	--
Total (%)	**100**	**100**	**100**

SAMPLE QUESTION (S)

Remember

1. Define machine learning
2. List the various similarity measures

Understand

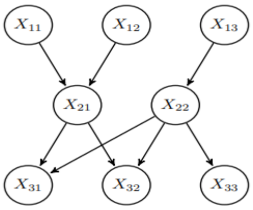
1. Explain the need of pre-processing in machine learning
2. Differentiate between supervised and unsupervised algorithms

Apply

1. Give Decision trees to represent the Boolean Functions:
   - **a.** A && - B
   - **b.** b. A V [B && C]
   - **c.** c. A XOR B
   - **d.** d. [A&&B] V [C&&D]

Open Book Question(s)

1. Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can 3 points that are assigned to different clusters in the kmeans solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example or explain in 1-2 sentences
2. Assume we have a set of data from patients who have visited UPMC hospital during the year 2011. A set of features (e.g., temperature, height) have been also extracted for each patient. Our goal is to decide whether a new visiting patient has any of diabetes, heart disease, or Alzheimer (a patient can have one or more of these diseases). Suggest a suitable machine learning approach for the above situation
3. Consider a Bayesian network B with boolean variables



a) Is there any variable(s) conditionally independent of X33 given X11 and X12? If so, list all
b) Is there any variable(s) conditionally independent of X33 given X22? If so, list all
c) Write the joint probability P(X11, X12, X13, X21, X22, X31, X32, X33) factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?
d) Write an expression for P(X13 = 0, X22 = 1, X33 = 0) in terms of the conditional probability distributions given in your answer to part (c).

21IT308 Database Management Systems Lab

0 0 3 1.5

Course Outcomes

At the end of the course, students will be able to

1. Demonstrate ER Modeling concepts to design the Database
2. Apply integrity constraints on a database
3. Make use of DDL, DML, DCL, TCL commands in creation and manipulation of Database
4. Utilize sub queries to make the complex queries more readable
5. Implementation of database queries using PL/SQL
6. Experiment with triggers to maintain the referential integrity of data

CO-PO Mapping

CO	PO4
1	3
2	3
3	3
4	3
5	3
6	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

List of Experiments

1. Design ER Model for a given application & Convert ER model to Relational Model.
2. Creating users - roles and Granting privileges.
3. Creating and altering tables for various relations in SQL using Integrity Constraints.
4. Implementing queries in SQL using
   - 4.1 Insertion
   - 4.2 Retrieval (operations like union - intersect – minus)
   - 4.3 Updating
   - 4.4 Deletion
5. Implement the following Sql operations in, any, all, between, like, exists , group by, Order by and having etc.
6. Implementing the concepts of Rollback – commit, checkpoints and Views
7. Implementing Different Types- Joins (Left, Right, Inner & Outer Joins)
8. Implementing Different Types - sub queries and nested queries.
9. Experiment with built in functions in oracle (Numeric, Aggregate functions etc.)
10. Implementing TO_CHAR(),TO_DATE(),TO_NUMBER() & String Handling Functions.
11. Implementing operations on relations using PL/SQL.
12. Implementing functions, stored procedures using PL/SQL
13. Implementing cursors using PL/SQL
14. Implement Exception Handling using PL/SQL
15. Creating triggers using PL/SQL
16. Creating Packages using PL/SQL

List of Augmented Experiments

1. Inventory control management System
2. College Management System
3. Hospital management System
4. Library management System
5. Payroll management System
6. Health care organization Management System
7. Restaurant Management System
8. Blood Donation Management System
9. Art Gallery Management System
10. Hotel Management System
11. School Management System
12. Salary Management System
13. Wholesale Management System
14. Time Table Management System
15. Website Management

Reading Material (s)

1. Database Management Systems Lab Manual, Department of CSE, GMRIT,Rajam

21DS407 Foundations of Data Science Lab 0 0 3 1.5

Course Outcomes

1. Demonstrate data formats and linear algebra matrices operation
2. Identifying similarities and dissimilarities between attributes
3. Importance data visualization by drawing plots
4. Demonstrate regression techniques on various datasets
5. Demonstrate Classification techniques on various datasets
6. Demonstrate Clustering techniques on various datasets

CO–PO Mapping

CO	PO4	PO5	PO8
1	3	2	2
2	3	3	2
3	3	1	3
4	2	2	2
5	2	2	3
6	1	2	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Experiments:

1. Linear Algebra Operations
2. Data Formats, Processing& Transformation
3. Obtaining Statistical Properties of data
4. Basic Experiments on Data Visualization, Interpretation, and Identification of outliers
5. Single Valued Decomposition and PCA
6. Optimization Techniques
7. Regression and variants (Prediction)
8. Regression with regularization (Prediction/classification)
9. Decision Tree Induction, Rule Generation and Classification
10. Implementation of K-Nearest Neighbor Classifier
11. Implementation of Bayesian Classifier
12. Implementation of Ensemble models
13. Implementation of K-Means Clustering
14. Implementation of Hierarchical clustering
15. Implementation of DBSCAN Clustering
16. Build, Train, Test and Deployment on cloud environment (e.g.AWS/Google Cloud)

Augmented Experiments:

Classification:
1. Design and Deploy a machine learning model for health care System
Sample Datasets:
a. https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-predictiondataset
b. https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000
2. Design and Develop a machine learning model for Business Analytics
Sample Datasets:
a. https://www.kaggle.com/datasets/arashnic/hr-analytics-job-change-of-data-scientists
3. Design and Develop a machine learning model for Education institutions
Sample Datasets:
a. https://www.kaggle.com/datasets/spscientist/students-performance-in-exams
4. Design and Develop a machine learning model for TV Shows
Sample Datasets:
a. https://www.kaggle.com/datasets/shivamb/netflix-shows
5. Design and Develop a machine learning model for E-Commerce applications

Sample Datasets:
a. https://www.kaggle.com/datasets/shivamb/netflix-shows
6. Design and Develop a machine learning model Supermarket store applications
Sample Datasets:
a. https://www.kaggle.com/datasets/surajjha101/stores-area-and-sales-data
Classification/Clustering:
7. Design and Develop a machine learning model for Drugs Review
Sample Datasets:
a. https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Druglib.com%29
8. Design and Develop a machine learning model for Plants
Sample Datasets:
a. https://archive.ics.uci.edu/ml/datasets/Plants
9. Design and Develop a machine learning model for TV News Channel Commercial Detection
Sample Datasets:
a. https://archive.ics.uci.edu/ml/datasets/TV+News+Channel+Commercial+Detection+Dataset
10. Design and Develop a machine learning model for SMS Spam Collection
Sample Datasets:
a. https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

21ESX01 Employability Skills I

0 0 2 2

Course Outcomes

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Introspect & develop life skills with constructive approach
3. Assess and improve analytical skills
4. Apply the Inter Process Communication in Linux environment
5. Develop a simple communication application to communicate using TCP/UDP client server model
6. Develop solutions to real-world problems

COs – POs Mapping

COs	PO1	PO2	PO5	PO8	PO10	PO12
CO1	-	-	-	-	3	-
CO2	-	-	-	-	2	2
CO3	2	-	-	2	-	-
CO4	3	-	2	2	-	-
CO5	3	2	2	-	-	2
CO6	2	2	2	-	-	-

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Unit-I

Communication Skills **16 hours**

Building Confidence: Fear? Steps to Overcoming the Fear of Public Speaking? Self Esteem: Definition? Types of Self Esteem, Causes of Low Self Esteem, Merits of Positive Self Esteem and Steps to build a positive Self Esteem.

Group Discussions (Practice): GD? GD Vs Debate, Overview of a GD , Skills assessed in a GD, Dos & Don'ts, & Conducting practice sessions (SimpleTopics).

Motivational Talk: Team Work: Team Vs Group? Stages in Team Building, Mistakes to avoid and Lessons to Learn (Through Stories or Can be a Case Specific)

Quantitative Aptitude **16 hours**

Percentages, Profit and loss, Mixtures and Allegations, Simple Interest, Compound Interest

Unit-II

Linux Programming **32 hours**

S.No	Topic	No.of Hours
1	**Understanding System Files** File Structures, System Calls for File Management – create, open, close, read, write, lseek, link, symlink, unlink, stat, fstat, lstat, chmod, chown,	4
2	**Directory API** opendir, readdir, closedir, mkdir, rmdir, umask.	2
3	**Industry Essentials** Network Configurations and Inter Process Communications, Signals, Pipes and Sockets	4
4	**Services of Linux** Running Services on Linux, KVM Virtualization, Logical Volume Machine, Tiger VNC, FTP Server, Yum Server	5

TEXT BOOKS:

1. W. Richard. Stevens (2013), Advanced Programming in the UNIX Environment, 3rd edition, Pearson Education, New Delhi, India.
2. Behrouz A. Forouzan, Richard F. Gilberg. Thomson (2003), Unix and shell Programming, Cengage Learning India

REFERENCES:

1. W. Richard Stevens, Andrew M. Rudoff, Bill Fenner (1998), UNIX Network Programming, PHI.
2. Graham Glass, King Ables (2006), UNIX for Programmers and Users, 3rd Edition, , Pearson Education

21HSX11 CC & EC Activities I

0 0 1 1

Course Outcomes

1. Interpret and present the abstractive technical information through an activity
2. Think critically in providing solutions to the generic and common problems
3. Demonstrate the creative thinking in dealing with liberal arts
4. Instill team sprit through active engagement with the peer
5. Develop programs of common interest having social impact
6. Empower the under privileged through motivational activities

COs –POs Mapping

COs	PO_6	PO_7	PO_9	PO_{10}
1				3
2	3	2		
3	3			
4			3	
5	3			
6	3			

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Co-Curricular and Extra Curricular (CCEC) Activities:

Students shall acquire 1 credit each in 2nd and 3rd years with the following scheme:
Scheme of evaluation for the CCEC activities:
- No. of slots in each Semester @ 1 slots every week : 12
- No. of Streams (1 CC + 1 EC + 1 Community Engagement) : 3
- No. of slots allotted for each stream : 4

Requirement for the award of 1-Credit
- Students shall choose at least two streams of events in each semester and among which Community engagement through NSS is mandatory
- Students shall secure an overall attendance of 75% in all events put together

The credits earned through these courses will be indicated in the grade memo and will be taken into account for CGPA calculation.

5th Semester

21IT405 Web Technologies

3 0 2 4

Course Outcomes

At the end of the course, students will be able to

1. Illustrate client-side scripting tools to create and modify web pages.
2. Infer the knowledge of web programming to develop and maintain web pages.
3. Develop portable and dynamic web pages.
4. Demonstrate server-side scripting languages to develop business logic.
5. Make use of database connectivity to communicate database server from web server.
6. Test for client-server web application development in real time internet domains.

CO–PO Mapping

COs	PO$_3$	PO$_5$	PSO$_1$
1	3	2	3
2	3	2	3
3	2	3	3
4	2	3	2
5	3	3	3
6	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Syllabus

Unit- I **12+8 Hours**

HTML Tags

Introduction, Links, Lists, Tables, Forms, Canvas, Semantics, Plug-ins.**CSS:** Introduction, CSS Properties, Selectors, Combinators, Controlling Fonts, Forms, Pseudo classes, Transitions, Animations (data-aos), 2D and 3D Transforms, Navigation Bar, Layouts,

Java Script: Introduction, Functions, Events, Validations, Objects, Document Object Model (DOM), Division Replacement, Browser Object Model (BOM)

CSS Tooltips, Font Combinations.

Practical Components

1. Design the following static web pages required for Online Examination System web site.
   a) Home Page
   b) Login Page: a. Student Login Page b. Admin Login Page
   c) Registration page
   d) Test Page
   e) Results Page.
2. For Specific pages of Online Examination System like Test page and Result page Apply different font styles, font families, font colors, animations and other formatting styles to the above static web pages.
3. Validate login page and registration page using Java Script.
4. Create an HTML page with any required JavaScript that takes a number from text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets and special characters.

Unit- II **12+8 Hours**

Bootstrap

Responsive Design, Layouts, Grids, Media Queries, **Components-** Forms, Drop Downs, Cards, Input Groups, Collapse, List Groups, Navbar, Popovers, Flex Box. **Utilities-** Borders, Positions, Scroll spy, Visibility, CSS Components, SVG, Java Script Components. **JSON:** Introduction, Parsing, Objects, JSON arrays, Dynamic HTML Data. **jQuery:** Selectors, Filters, Form Plug-ins, Auto Validations

Bootstrap icons, Jumbotron.

Practical Components

1. Design an interactive area using HTML and CSS. Use JavaScript to implement the color change effect during mouseover and reversion during mouse out.
2. **Inner Function with Constant Addition**: Design a function "createAdderWithConstant" that takes a constant as a parameter and returns an inner function. The inner function, when given a number, adds the constant to it and returns the result. Use the inner function to add the same constant to different numbers.
3. Make the static pages Responsive and attractive using Bootstrap components (***Mobile View)
4. Validate login page and registration page using JQuery**.**

Unit- III **12+8 Hours**

PHP

Introduction, Modal View Architecture (MVC), Creating PHP script, Running PHP script, Introduction to PHP, Arrays, Functions, sending parameters in URL, working with Forms, **working with Database:** Running SQL Queries, connecting with Databases, Uploading Files, PHP Sessions, Redirecting, Routing, Connecting PHP with JavaScript and jQuery, Authentication.

PHP Cookies, PHP Exception Handling, Introduction to PHP Laravel

Practical Components

1. Write a PHP to connect to the database, Insert the details of the student who registered through Online Examination System student login page including photograph.
2. Develop and demonstrate PHP Script for the following problems:
   a. Write a PHP Script to find out the Sum of the Individual Digits.
   b. Write a PHP Script to check whether the given number is Palindrome or not
3. Write a PHP program to insert the questions and respective answers into the database through admin page, sending mails to registered students through admin.
4. Write a PHP program, assuming four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Create a Cookie and add these four user ID‟s and passwords to this Cookie. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.
   a. If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display "You are not an authenticated user"

Unit- IV **12+8 Hours**

Flask

Introduction to Flask, Virtual Environment, features of flask, url building, routing, Templates and Jinja Code, Rendering Templates, Static files, Building Forms, Sending Form data to Templates, Template Inheritance (header, footer, etc), Session Tracking, connecting database (SQLite), Retrieving database values to templates, File uploading, Sending Mails, Deployment of website.

wsgi file, csrf token, http methods.

Practical Components

1. Write a Flask code to connect to the database, and authenticate Login pages with jQuery and start a Session for Student, Change password for Student
2. After user login, display Questions from the database into test page in shuffled manner using flask and store in database.
3. Write a Flask code to do the following

a) Evaluate the answers of the test which is given by the student in the Test page.
b) Calculate the total score of the student and store it into the database and display score in dashboard of student.

4. Using flask retrieve results in a responsive table format of all students with filtering. Deploy the application developed in real time environment

Total: 48+32 Hours

Textbook(s):

1. Programming the World Wide Web, 8th edition Robert W. Sebesta, pearson.
2. Bootstrap: Responsive Web Development, 1st Edition, jake spurlock foreword by dave winer, O'Reilly publications.

Reference Book(s):

1. Web programming with HTML, XHTML and CSS, 2e, Jon Duckett, Wiley India
2. Web programming Bai, Michael Ekedahl, CENAGE Learning, India edition.
3. An Introduction to Web Design + Programming, Paul S.Wang, India Edition
4. Flask Web Development 2e Miquel Grinberg O'Reilly publications.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Lab Examination (%)
Remember	20	20	--
Understand	50	40	--
Apply	30	40	40
Analyze	--	--	20
Evaluate	--	--	40
Create	--	--	--
Total (%)	100	100	100

Sample Question (S)

Understand

1. What is the use of cascading style sheet? Explain different style sheets used in HTML.
2. Explain string object with suitable example.
3. Illustrate how work java script events.
4. Describe procedure for how to connect PHP to MYSQL.
5. Explain AJAX technologies

Apply

1. Write an HTML program to display your class timetable.
2. Write a java script to find reverse of a given number using functions.
3. Design a responsive registration page using bootstrap.
4. Insert an image into a web page. Write a script which displays a message when the mouse is over the image. The co-ordinates of the mouse should be displayed if click is attempted on the image.
5. A person X has created the table (emp) and inserted the data in the table. After the completion of insertion, he found that one of the inserted record is invalid. So, write a PHP program to update the table by deleting the invalid record. [Assume database table name is emp (empno, ename, dept, age)]

Analyse

1. Compare traditional web application development with AJAX and analyze what way AJAX useful for real time internet application.

21DS502 Deep Learning for Data Science

3 0 0 3

Course Outcomes

1. Illustrate the knowledge of the basic Concepts of Neural Networks.
2. Classify the various types of Learning rules.
3. Understand the concept of Single and Multi-Layer Perception.
4. Make use of different CNN models.
5. Compare different character encoding techniques.
6. Exemplify different Deep unsupervised models.

COs–POs Mapping

COs	PO1	PO2	PO4	PO5	PO12
1	3	2	2	1	1
2	3	2	2	1	1
3	3	3	3	1	1
4	3	3	3	1	2
5	3	3	3	3	2
6	3	3	3	3	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12 Hours**

Introduction: Introduction to Deep Learning, Historical Trends in Deep Learning, The Neural Network, limitations of traditional computer programs, The Mechanics of Machine Learning, The Neuron, Expressing Linear Perceptron as Neurons, Feed-Forward Neural Networks, Linear Neurons and TheirLimitations, Types of Activation Function: Sigmoid, Tanh, ReLU, and Softmax, learningrules

Bias, Mean, Variance, Co-Variance

Unit II **12 Hours**

Single Layer Perceptron: Adaptive Filtering Problem, Unconstrained Organization Techniques-Method of Steepest Descent, Newton's Method, Gauss Newton Method, Perceptron-Convergence Theorem, Relation Between Perceptron, Bayes Classifier for a Gaussian Environment

Multilayer Perceptron: Back Propagation Algorithm-Two Passes of Computation, Activation Function, Rate ofLearning, Sequential and Batch Model of Training, Stopping Criteria, optimization settings.

Linear Least Square Filters, Least Mean Square Algorithm, XOR Problem

Unit III **12 Hours**

convolution networks: Building blocks of CNNs, Architectures, Filters and Feature Maps, pooling layers, Convolutions over volumes, Softmax regression, Deep Learning frameworks, Training and testing on different distributions, Bias and Variance with mismatched data distributions, Transfer learning, Multi-tasklearning, end-to-end deep learning.

CNN models: AlexNet, VGG –16, Residual Networks, YOLO

Keras ,tensorflow, data augmentation, Batch Normalization, Dropout

Unit IV **12 Hours**

Recurrent Networks

One-hot encoding of words and charcters, using word embedings, Recurrent Neural Network Model, Vanishing gradients with RNNs, Gated Recurrent Unit (GRU), LSTM (long short term memory), Encoder Decoder sequence to sequence architectures.

Deep Unsupervised Learning: Autoencoders, variational Autoencoders, Generative adversarial network.

n-grams ,bag-of-words, Bi directional RNN, Exploding gradient

Total: 48 Hours

Textbook (s)

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron, October 2022: Third Edition.
2. Simon Haykin, "Neural Networks - A comprehensive foundation", Pearson Education, 2003.
3. S. N. Sivanandam, S. N. Deepa, "Principles of Soft Computing", Willy, 3rd edition, 2019.
4. Nikhil Buduma, "Fundamentals of Deep Learning", ORELLIY, 1st Edition, 2017.
5. Laurene Fausett, "Fundamentals of Neural Networks" , Pearson Education, 2004.
6. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep Learning." An MIT Press book in preparation,2015

Reference (s)

1. Satish Kumar,"Neural Networks: A Classroom Approach" Tata McGraw Hill Education, 2004.
2. Simon Haykin, " Neural networks and Learning Machines", Prentice Hall, 2008
3. Josh Patterson and Adam Gibson "Deep Learning A Practitioner's Approach" O'Reilly Media, Inc. 2017

SAMPLE QUESTION (S)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Lab Examination %
Remember	20	20	---
Understand	40	40	---
Apply	40	40	40
Analyze	---	---	40
Evaluate	---	---	20
Create	---	---	---
Total (%)	100	100	100

Remember

1. Define Neural Networks.
2. Write Short notes on Hebbian Learning.
3. What is mean by Perceptron
4. What are the applications of Deep Learning?

Understand

1. What is are the benefits of Neural Networks?
2. How can deep learning models be optimized?
3. How do you evaluate deep learning models?
4. What are some common architectures used in deep learning?

Apply

1. Discuss some popular deep learning frameworks and libraries?
2. Discuss various activation function in CNN models

Analyze

1. Illustrate the working model of VGG-16 with diagram

21DS503 Data Analytics and Visualization Techniques

3 0 2 4

Course Outcomes

1. Explain the Data Analysis Fundamentals
2. Illustrate various data Visual aids
3. Categorize different data transformation and descriptive statistics
4. Identify different Correlation and Inferences from statistical tests
5. Make use of the concept Hypothesis Testing
6. Classify different Multivariate Analysis techniques

COs–POs Mapping

COs	PO1	PO4	PSO1
1	3	2	3
2	3	3	3
3	3	2	3
4	2	3	3
5	3	3	3
6	3	2	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12+8 Hours**

Data Analysis Fundamentals

Understanding data science, significance of EDA, steps in EDA. types of analysis (univariate, bivariate, multivariate). Making sense of data: Numerical data-Discrete &continuous data, categorical data, Measurment scales-Nominal, Ordinal, Interval, Ratio. Comparing EDA with classical and Bayesian analysis, getting started with EDA: Numpy, Pandas,Scipy, and Matplotlib.

Grouping of data: Groupby mechanics, rearranging, reshaping data structures, data aggregation methods, and cross-tabulation methods.

Objectives of Exploratory data Analysis, The applications of EDA

Practical Components

Experiment 1: Explore Scipy Library
Experiment 2: Explore Stat Library
Experiment 3: Experiment with all group by functionalities & cross tabulation methods
Experiment 4: Analyzing and Visualizing Real-World Data.

Unit II **12+8 Hours**

Data transformation and descriptive statistics

Transformation techniques- performing data deduplication, replacing values, handling missing data, renaming axis indexes, outlier detection and filtering. Permutation and random sampling, computing indicators/dummy variables, string manipulations.

Correlation and Inferences: Introducing correlation, covariance, Pearson's Correlation, Spearman's Rank Correlation.

Descriptive statistics: Understanding statistics, measure of central tendenacy, measure of dispersion (standard deviation, variance, skewness, kurtosis, percentiles, quartiles)

Violin Plots, Inter Quartile Range, Discretization & binning, heatmaps

Practical Components

Experiment 5: Experiment with various data pre-processing & Transformation techniques
Experiment 6:Experiment and Analyse the correlations among attributes.
Experiment 7: Experiment with the statistical measures
Experiment 8: Outlier Detection and Handling.

Unit III **12+8 Hours**

Visualizing Distributions

Visualizing Amounts-Bar Plots, Grouped and Stacked Bars, Dot Plots and Heatmaps, Visualizing

Distributions: Histograms and Density Plots- Visualizing a Single Distribution, Visualizing Multiple Distributions at the Same Time, Visualizing Distributions: Empirical Cumulative Distribution Functions and Q-Q Plots-Empirical Cumulative Distribution Functions, Highly Skewed Distributions, Quantile Plots, Visualizing Many Distributions at Once-Visualizing Distributions Along the Vertical Axis, Visualizing Distributions Along the Horizontal Axis

Kernel Density Estimation (KDE), Geospatial Distributions, Interactive Visualization

***Practical* Components**

Download the House Pricing dataset from Kaggle

Experiment 9: Use different Color scales on the Rainfall Prediction dataset
Experiment 10: Create different Bar plots for variables in any dataset
Experiment 11: Visualization of different Distributions in various ways
Experiment 12: Visualizing Empirical Cumulative Distribution Functions (ECDF) for Real-world Data.

Unit IV **12+8 Hours**

Visualizing Associations & Time Series

Stacked Densities, Visualizing Proportions Separately as Parts of the Total ,Visualizing Nested Proportions-Nested Proportions Gone Wrong, Mosaic Plots and Treemaps, Nested Pies ,Parallel Sets. Visualizing Associations Among Two or More Quantitative Variables-Scatterplots, Correlograms, Dimension Reduction, Paired Data. Fundamentals on Time Series Data, Visualizing Time Series and Other Functions of an Independent Variable-Individual Time Series , Multiple Time Series and Dose–Response Curves, Time Series of Two or More Response Variables

Multidimensional Scaling, MANOVA vs ANOVA

Practical Components

Experiment 13: Visualization through Pie charts, Stacked Bars & Stacked Densities
Experiment 14: Show an example of Skewed data and Experiment with different skewness removal techniques
Experiment 15: For a sales dataset do a Time Series Visualization
Experiment 16: Geospatial Visualization for Regional Sales Analysis

Total: 45+32 Hours

Textbook (s)

1. Daniel J. Denis: Univariate, Bivariate, and Multivariate Statistics Using R: Quantitative Tools for Data Analysis and Data Science, Wiley,2020
2. Mukhiya Suresh Kumar Mukhiya, Ahmed Usman Ahmed: Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data, Packt, 2020
3. Claus Wilke, “Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures”, 1st edition, O’Reilly Media Inc, 2019.
4. Downey, Allen. Think stats: exploratory data analysis. " O'Reilly Media, Inc.", 2014.
5. Neil H. Spencer: Essentials of Multivariate Data Analysis, CRC Press,2014

Reference (s)

1. Wes McKinney : Python for Data Analysis 2nd Edition,Wiley,2013
2. Glenn J. Myatt, Wayne P. Johnson: Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining, 2nd Edition,Wiley,2014
3. 1 Tony Fischetti, Brett Lantz, R: Data Analysis and Visualization,O’Reilly ,2016 2 OssamaEmbarak, Data Analysis and Visualization Using Python: Analyze Data to Create Visualizations for BI Systems,Apress, 2018
4. Wendy L. MartinezAngel R. MartinezJeffrey L. Solka: Exploratory Data Analysis with MATLAB, 2nd Edition,CRC Press,2011
5. Radhika Datar, Harish Garg : Hands-On Exploratory Data Analysis with R , Packt,2020
6. Joseph F Hair, Barry J. Babin, Rolph E. Anderson, William C. Black: Multivariate Data Analysis Cengage, 2018

SAMPLE QUESTION (S)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	50	40	--
Understand	30	40	--
Apply	20	20	50
Analyze	--	--	50
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

Remember

1. What is data science.
2. Define purpose of different data visualization aids
3. List any two steps involved in EDA.

Understand

1. Explain different data representation
2. Explain the role of various data grouping methods
3. Explain purpose of hypothesis testing in detail

Apply

1. Apply various visual aids to identify behavior of data
2. Make use of Multivariate Analysis methods for data nanlysis
3. Examine data with various distribution function

Analyze

1. Compare and contrast various data grouping methods.
2. Distinguish various methods to address problems with dummy variables
3. Examine various string handling methods

Evaluate

1. Evaluate the statistics of a given dataset
2. Justify the importance of data skewness
3. Measure data dispersion methods on a dataset

Open Book Exam Questions

Question 1:

Perform below analysis on a house price prediction dataset.

a) Univariate Analysis
   i. Analysis of a numerical feature
   ii. Analysis of a categorical feature
b) Bivariate Analysis
   i. Relationship of a numerical feature with another numerical feature
   ii. Relationship of a numerical feature with a categorical feature

Question 2:

Perform below analysis on weather prediction dataset.

1. Correlation Analysis
   i. Correlation Heat Map
   ii. Zoomed Heat Map
2. Investigation of missing values
   1. What's missing? to what extent?
   2. Visualizing missing values in a DataFrame

21ML504 Computer Networks

3 0 0 3

Course Outcomes

1. Understand the basic network infrastructure to learn the overall functions of networking systems
2. Classify various wired and wireless transmission media for data communication networks
3. Apply knowledge of different techniques of error detection and correction to detect and solve error bit during data transmission.
4. Compare various routing algorithm and select an appropriate one for a routing design.
5. Design a proper routing for IP networks.
6. Understand the internal functionalities of main protocols such as HTTP, FTP, SMTP, TCP, UDP, IP.

CO-PO Mapping:

COs	PO1	PO2	PO3
1	3	2	1
2	3	3	1
3	1	3	2
4	2	3	1
5	1	3	3
6	3	1	1

3- Strongly linked, 2- Moderately linked, 1- Weakly linked

SYLLABUS:

UNIT I **12 Hours**

Network Hardware-Network Software-and Transmission Media

OSI-TCP/IP reference models, ARPANET, Network Topologies,

Physical Layer: Transmission media: Guided Media-Twisted Pair-Coaxial Cable-Fiber optics-Unguided Media: Electromagnetic Spectrum, Radio Transmission and Microwave Transmission. Switching Techniques: Circuit Switching-Packet Switching-Message Switching.

Taxonomy of networking devices.

Unit II Data link layer

12 Hours

Design Issues: Framing-error detection and correction-CRC-Elementary Data link Protocols: Stop and wait-Sliding Window protocols: Go-back-n-Selective Repeat ARQ

Medium Access sub layer and Multiple Access protocols: ALOHA-CSMA-IEEE Standard 802.3 and Ethernet-IEEE Standard 802.4: Token bus.

Data Link Control Protocols: HDLC-SLIP-PPP

Unit III Network Layer

12 Hours

Network Layer design issues-Virtual circuit and Datagram Subnets-Routing algorithms: shortest path routing-Flooding-Hierarchical Routing-Distance vector routing-Broad cast and Multi cast routing, Congestion Control: Congestion prevention policies.

Internet control protocols

Unit IV Transport Layer

12 Hours

Transport Services-Connection management - Elements of Transport Protocols-Internet Transport Protocols: UDP and TCP. IPv4 & IPv6, Application Layer-Domain name system - Electronic Mail-WWW

ATM Transport protocol-Proxy Servers-Multimedia: Data compression-digital representation of analog signals

48 Hours

Textbook (s)

1. Andrew S Tanenbaum, Computer Networks, 4th Edition, Pearson Education /PHI, 2002.
2. Behrouz A. Forouzan, Data Communications and Networking, 3rdEdition, Tata McGraw Hill Higher Education, 2003

Reference (s)

1. Willam Stallings, Data and Computer Communications,8th Edition, Pearson Prentice Hall, 2007.
2. W.A. Shay, Thomson, Understanding communications and Networks, 3rd Edition, Cengage Learning, 2005

SAMPLE QUESTION (S)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	25	20	--
Understand	40	40	20
Apply	30	40	40
Analyze	--	--	40
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

Sample Questions

Remember

1. List the applications of Computer Networks
2. Write two differences between OSI and TCP/IP models.
3. State 5 key assumptions in Dynamic channel allocation?
4. State the purpose of DNS.
5. Define congestion.

Understand

1. Illustrate OSI Reference model.
2. Explain various design issues of data link layer
3. What are the responsibilities of Data Link layer and explain Pure Aloha and Slotted Aloha protocols.
4. Describe Distance Vector routing algorithm with example and explain count to infinity problem 5. Represent the Manchester encoding for the bit stream: 0001110101.

Apply

1. What is the check summed frame transmitted if the message is 1101011011 and the generator polynomial is x 4 + x + 1 using CRC
2. Can you think of any circumstances under which an open-loop protocol, (e.g., a Hamming code) might be preferable to the feedback-type protocols?
3. Assuming that all routers and hosts are working properly and that all software in both is free of all errors, is there any chance, however small, that a packet will be delivered to the wrong destination?
4. The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used: A B ESC C ESC FLAG FLAG D. What is the output after stuffing?
5. The following character encoding is used in a data link protocol: A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000 Show the bit sequence transmitted (in binary) for the four-character frame: A B ESC FLAG when each of the following framing methods are used:
    1. Character count.
    2. Flag bytes with byte stuffing.
    3. Starting and ending flag bytes, with bit stuffing.

21MLC11 Computer Vision & Pattern Recognition

3 0 0 3

Course Outcomes

1. Explain the fundamental concepts and principles of computer vision, including its applications and challenges.
2. Apply various point operators, linear filtering, and neighborhood operators to perform basic image processing tasks
3. Explain the principles, algorithms, and techniques used in feature detection, matching, and segmentation.
4. Describe the concepts of pattern recognition and its applications.
5. Understand and apply supervised and unsupervised learning methods to discover patterns and group similar data points.
6. Understand and apply Recognition and object detection techniques to locate and identify objects in images or video sequences.

CO–PO Mapping

CO	PO1	PO3	PSO1	PSO2
1	3	2	2	2
2	2	3	3	3
3	3	2	2	2
4	3	2	2	2
5	2	3	3	3
6	2	3	3	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12 Hours**

Introduction to Computer Vision and Image Processing

Computer Vision: Overview and introduction - Image formation: Geometric primitives, transformations, and photometric image formation - Sampling and aliasing in images - Image processing: Point operators, linear filtering, neighborhood operators - Multi-resolution representations and wavelets.

Image registration and content-based image retrieval.

Unit II **12 Hours**

Feature Detection, Matching, and Segmentation

Feature detection and matching: Points and patches – Feature detector, feature descriptors, feature Matching, Edges - Edge detection and linking. Lines - Piecewise linear, Hough transform. Segmentation techniques: Active contours (snakes), split and merge, region splitting, region merging - Mean shift and mode finding and K-means based segmentation.

Graph cuts and energy-based methods, Medical image segmentation.

Unit III **12 Hours**

Pattern Recognition and Machine Learning

Introduction to pattern recognition - Training and learning in PR systems- Different approaches to pattern recognition - Statistical pattern recognition - Supervised learning with parametric and nonparametric methods - Unsupervised learning and clustering - Neural pattern recognition: Neural network structure for pattern recognition applications, feed-forward neural networks.

Dimensionality reduction methods- PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis)

Unit IV **12 Hours**

Recognition, Tracking, and Motion Estimation

Object detection - Face recognition - Instance recognition –Image Classification–Semantic segmentation - Tracking and motion estimation: Optical flow, activity recognition, motion estimation, and tracking.

Deep learning object detection using CNNs, YOLO, and SSD

Total: 48 Hours

Textbook (s)

1. Richard Szeliski, Computer Vision: Algorithms and Applications , Springer, 2022, ISBN:978-1848829343
2. Rober.J. Shelkoff, John Wiley &Sons,Pattern Recognition- Statistical, Structural and Neural Approaches,Wiley, 2007, ISBN: 978-8126513703.

Reference (s)

1. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, Wiley India, 2006, ISBN: 978-8126511167.
2. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012,78-1107011793
3. Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis, and Machine Vision, Brooks/Cole, 2007, ISBN:978-0495082521.

Internal Assessment Pattern

Cognitive Level	**Int. Test 1 (%)**	**Int. Test 2 (%)**	**Open Book Exam**
Remember	30	30	-
Understand	50	50	-
Apply	20	20	50
Analyze	--	--	50
Evaluate	--	--	-
Create	--	--	-
Total (%)	**100**	**100**	**100**

Remember

1. What is computer vision and what is its role in image processing?
2. Define Hough transform.
3. List the roles of linear filtering in image processing.
4. What is Instance recognition?

Understand

1. How does sampling affect the quality of an image?
2. Explain the working principles of K-nearest neighbors (KNN) and decision trees in pattern recognition.
3. Explain the concept of wavelets and their use in multi-resolution representations.
4. Describe the process of feature detection in computer vision.
5. Describe the process of optical flow estimation and its applications in motion tracking.

Apply

1. Discuss the application of mean shift and mode finding in image segmentation.
2. Explain the concept of multi-resolution representations in image analysis. How are wavelets used in this context?

Analyze

1. Analyze the impact of aliasing in images and discuss methods to mitigate its effects.
2. Compare and contrast parametric and nonparametric methods in supervised learning for pattern recognition.
3. Analyze the challenges and potential solutions for object tracking in crowded or occluded scenes.

21CSC21 Web Programming Languages

3 0 0 3

Course Outcomes

1. Understand web applications.
2. Analyze the templates and common scenarios of web development.
3. Examine various web programming languages and their usages.
4. Analyze the flow of data through various layers of web.
5. Identify the use of data management and compatibility of web browsers.
6. Identify the need of security and performance for a web application.

COs–POs Mapping

COs	**PO1**	**PO2**	**PO7**	**PO12**
1	3	3	2	2
2	3	3	2	2
3	3	3	2	2
4	3	3	2	2
5	3	3	1	2
6	3	1	1	1

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12 Hours**

Overview of the web

Web application Overview: Basic concepts of web, Hyper Text Mark-up Language (HTML) and Cascading style sheets (CSS), Web programming languages
Web server: Introduction, Anatomy of HTTP transaction, Request and response structures, REST APIs. Data management: Browser storage **-** Local storage & Session storage, Database storage
HTML, CSS, HTTP, Status codes

Unit II

12 Hours

Web programming in JavaScript

Introduction: Basic concepts – Arrays, objects, functions, classes and interfaces, ES6 JavaScript standard. Node JS: Introduction, Node packages, HTTP web server – Creating a HTTP server that supports Create, Read, Update and Delete operations via various HTTP methods, Adding common request parser

Express: Creating a web server with express and body parser packages

NodeJS, ES6, Node packages.

Unit III **12 Hours**
Introduction: Basic concepts – Arrays, objects, methods, classes and interfaces, Extensible Markup Language (XML).
Web programming techniques – Servlets, Java Server Pages (JSP)
Servlets: Introduction, creating a HTTP servlet for Create, Read, Update and Delete operations
Java Server Pages (JSP): Introduction, creating a JSP for Create, Read, Update and Delete operations
Servlet, XML, JSP

Unit IV **12 Hours**
Web programming in Python
Introduction: Basic concepts – Arrays, objects, methods, classes and interfaces, Python's HTTP package**HTTP server**: Introduction, creating a server that supports Create, Read, Update and Delete operations*HTTP server, Python packages*

Total: 48 Hours

Textbook (s)

1. Learn Web Development with Python, Fabrizio Romano, Gaston C. Hillar, Arun Ravindran, 1st Edition, 2018.

2. Learn Java for Web Development, Vishal Layka, 1st Edition, 2014.
3. Node.js, John Bach, Alexander Aronowitz, 3rd Edition, 2021

Reference (s)

1. HTML 5 in simple steps, Kogent Learning Solutions Inc, Dreamtech Press.
2. Beginning HTML, XHTML, CSS, and JavaScript, John Duckett, Wiley India.
3. Beginning CSS: Cascading Style Sheets for Web Design, Ian Pouncey, Richard York, Wiley India.
4. Web Designing & Architecture-Educational Technology Centre, University of Buffalo.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	20	20	--
Understand	30	10	--
Apply	30	30	40
Analyze	10	20	30
Evaluate	10	20	30
Create	--	--	--
Total (%)	100	100	100

Sample Question(s)

Remember

1. List any four HTML tags and CSS properties
2. Define web server
3. List four browser storage methods

Understand

1. What HTTP method should be used for updating data in a server?
2. Explain the of CSS in web development
3. Illustrate the working of a HTTP server

Apply

1. When user opens a website, the underlying web application wants to store browser specific data.What is a good approach for saving the data?
2. Give an example of a scenario that might benefit from browser storage instead of backendstorage.

Analyze

1. There is a common body parsing workflow that every request to a server must obey. What is agood way to implement it?
2. Compare Servlet and Java Server Page
3. Analyze the general strategy behind cookies and session management.

Evaluate

3. Assuming the server received a permission error from the database, what is the best way to propagate it to the user?
4. Describe how to implement user authentication in a web server.

Open Book Exam Questions

1. A file is to be stored in the web server. However, the access to this file is to be limited to specificset of users. How can once achieve this using a web server?

2. Consider a request that yields data of high cardinality. How can one send the data withoutaffecting the performance of the website?

21MLC31 Fundamentals of Cloud Computing

3 0 0 3

Course Outcomes

1. Interpret the architecture and infrastructure models of cloud computing, strengths, and limitations of cloud computing.
2. Understand the virtualization concepts of machines and datacenters.
3. Infer the design concepts of cloud ready applications
4. Compare different cloud centre's implementation
5. Understand the concepts of cloud scaling and disaster recovery
6. Interpret the security and risk issues in cloud computing

COs–POs Mapping

COs	PO$_2$	PO$_6$	PO$_7$	PO$_8$
1	3	2	3	1
2	3	1	2	3
3	2	3	1	2
4	2	3	3	1
5	2	3	1	2
6	2	3	3	3

3–Stronglylinked|2–Moderatelylinked|1–Weaklylinked

Unit I **12 Hours**

Cloud Computing

Cloud computing: Introduction, SOA, Cloud computing architectures, Value of cloud computing, Cloud Infrastructure models including SaaS, PaaS, IaaS, Storage-as-a-Service, Database-as-a-Service, Information-as-a-Service,Process-as-a-Service,Application-as-a-Service, Integration-as-a-Service, and Security-as-a-Service. Public cloud, private cloud and hybrid cloud, Cloud Services.

Before moving into the cloud: Know Your Software Licenses, The Shift to a Cloud Cost Model. History of Cloud Computing, Advantages of Cloud Computing, and Disadvantages of Cloud Computing.

Companies in the Cloud Today, Amazon Web Services, Google services, IBM Cloud, Windows Azure, Tata Cloud, Salesforce.com

Unit II **13 Hours**

Virtualization & Design

Virtualization,Virtualmachine, Implementation Levels of Virtualization,Virtualization Structures/Tools and Mechanisms, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management, Datacentre, Virtualization for Data-Centre Automation. Service Levels for Cloud Applications Ready for the cloud: Web Application Design, Machine Image Design, Privacy Design, Database Management.

Various hypervisors like VMware, KVM,oracleVM,

Unit III **12 Hours**

Cloud centres, Scaling a Cloud Infrastructure and Disaster Recovery:

Cloud centres in detail: Comparing approaches, Xen, Eucalyptus, CloudStack, and OpenStack. Cloud Scaling:Capacity Planning, Cloud Scale, Types of cloud scaling.

Disaster Recovery:Disasters in cloud,Disaster Recovery Planning, Cloud Disaster Management.

Requirements for modern data centres-high availability and Service Orientated-Infrastructures(SOI).Modern data centre use case studies.

Unit IV **11 Hours**

Cloud Computing Software Security Fundamentals

Data Security, Network Security, Host Security, Compromise Response.

Cloud information Security Objectives, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Approaches to Cloud Software Requirements Engineering, Cloud Security Policy Implementation.

Cloud Computing Risk Issues: The CIA Triad, Privacy and Compliance Risks, Threats to Infrastructure Data and Access Control, Cloud Access Control Issues, Cloud Service Provider Risks.

MicrosoftAzure, IBMBluemix

Total:48 Hours

Textbook(s)

1. GeorgeReese,Cloud ApplicationArchitectures,1stEdition O'ReillyMedia,2009
2. RonaldL.KrutzandRussell DeanVines,CloudSecurity,1stEdition,WileyPublishing,2010

Reference(s)

1. MichaelMiller,CloudComputing-WebBasedApplicationsthatchangethewayyouworkandcollaborateonline, 1st Edition,PearsonEducation,Publishing, 2011
2. KaiHwang,GeoffreyCFoxandJackJ.Dongarra,Distributed&CloudComputingfromParallelProcessingtotheInternet ofThings,1stEdition, MK Publishing,2010
3. DavidSLinthicum,CloudComputingandSOAConvergenceinYourEnterprise:AStep-by-StepGuide,1stEdition, Addison-Wesley, 2009

SAMPLEQUESTION(S)

Internal Assessment Pattern

Cognitive Level	Int.Test1 (%)	Int.Test2 (%)	AssignmentTest3(%)
Remember	40	45	20
Understand	40	45	60
Apply	20	10	
Analyze	--	--	20
Evaluate	--	--	--
Create	--	--	--
Total(%)	100	100	100

Remember

1. Define Cloud Computing.
2. List types of virtualization.
3. Define pro active scaling.
4. What is CIAT RAID?

Understand

1. Explain different cloud Infrastructure models.
2. Explain different levels of virtualization.
3. Explain about Recovery Point Objective.
4. With neat diagram explain the functioning of Xen Architecture.
5. List and explain various cloud service providers risks.

Analyze

1. Compare cloud center and service infrastructure.
2. Analyze different cloud services provided by Amazon

21CS004 Principles of Programming Languages

3 0 0 3

Course Outcomes

1. Explain the concepts of programming languages.
2. Describe syntax and semantics of programming languages.
3. Explain data types, and basic statements of programming languages.
4. Illustrate the concept of subprogram.
5. Demonstrate the concepts of ADT and OOP.
6. Summarize the concept of exception handling and event handling.

CO–PO Mapping

COs	PO1	PO2	PO3	PO4
1	3	1	2	2
2	3	2	1	2
3	3	1	3	1
4	3	2	2	2
5	3	2	2	3
6	3	2	2	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I

Preliminary Concepts **12 Hours**

Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, language design trade-offs, implementation methods, programming environments, Evolution of Major Programming Languages.

Syntax and Semantics: General problem of describing syntax, formal methods of describing syntax.

attribute grammars, describing the meanings of programs.

Unit II

Data types, Expressions and Statements, Control Structure **12 Hours**

Names, Bindings, and Scopes: Introduction, names, variables, concept of binding, scope, scope and lifetime, referencing environments, named constants

Data types: Introduction, primitive, character, string types, user defined ordinal types, array, associative arrays, record, tuple types, list types, union types, pointer and reference types, type checking, strong typing, type equivalence

Expressions and Statements: Arithmetic expressions, overloaded operators, type conversions, relational and Boolean expressions, short- circuit evaluation, assignment statements, mixed-mode assignment

Control Structures: introduction, selection statements, iterative statements, unconditional branching.

Guarded commands.

Unit III

Subprograms **12 Hours**

Subprograms: Fundamentals of subprograms, design issues for subprograms, local referencing environments, parameter passing methods, parameters that are subprograms, calling subprograms indirectly, overloaded subprograms, generic subprograms, design issues for functions, user defined overloaded operators, closures, co routines

Implementing subprograms: General semantics of calls and returns, implementing simple subprograms, implementing subprograms with stack-dynamic local variables, nested subprograms, blocks.

Implementing dynamic scoping.

Unit IV

ADT, OOP, Exception and Event Handling **12 Hours**

Abstract Data types: The concept of abstraction, introductions to data abstraction, design issues, language examples, parameterized ADT, encapsulation constructs, naming encapsulations

Object Oriented Programming: Design issues for OOP, OOP in Smalltalk, C++, Java, Ada 95, C#, Implementation of Object-Oriented constructs.

Exception Handling and Event Handling: Introduction, exception handling in Ada, C++, Java.

Introduction to event handling, event handling with Java and C#.

Total: 48 Hours

Textbook (s)

1. Robert W. Sebesta, "Concepts of Programming Languages", Tenth Edition, Addison Wesley, 2012.
2. Programming Languages, Principles & Paradigms, 2ed, Allen B Tucker, Robert E Noonan, TMH

Reference (s)

1. R. Kent Dybvig, "The Scheme programming language", Fourth Edition, MIT Press, 2009.
2. Jeffrey D. Ullman, "Elements of ML programming", Second Edition, Prentice Hall, 2098.
3. Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009.
4. W. F. Clocksin and C. S. Mellish, "Programming in Prolog: Using the ISO Standard", Fifth Edition, Springer, 2003

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	50	40	--
Understand	50	50	--
Apply	--	10	50
Analyze	--	--	50
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. Define programming language?
2. Define named constants?
3. Define subprogram?

Understand

1. Explain numeric types?
2. Summarize grammars for simple assignment statements.
3. Discuss language evaluation criteria and the characteristics that affect them.

Apply

1. Identity the exceptions occurred in C++ and Ada.
2. Construct BNF notation for following:
   a. For loop
   b. If-else condition
   c. Structure definition
3. Solve the given grammar is ambiguous or not.

```
<assign> -><id>=<expr>
<id>->A|B|C
<expr>-><expr>+<expr>
|<expr>*<expr>
|(<expr>)
|<id>
```

21CS005 Mobile Computing

3 0 0 3

Course Outcomes

1. Explain the basic concepts and fundamentals of mobile computing and telecommunication systems along with various standards.
2. Illustrate the techniques, protocols related to GSM and GPRS architecture to perform requirements analysis.
3. Explain major components of Mobile IP to improve the service qualities of a network
4. Compare various ad hoc routing protocols to examine the performance of network
5. Explain the architecture of Wireless Sensor Network and WLAN design issues and limitations.
6. Outline the basic knowledge in developing smart phone applications using various platforms, toolkits, APIs and third party libraries

CO–PO Mapping

COs	PO3	PO8
1	3	2
2	2	2
3	1	2
4	2	2
5	2	2
6	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I

Introduction to Mobile Communications **12 Hours**

Overview: Mobile Communication and Mobile Computing – Architecture; Generations of Telecommunications – 1G, 2G, 3G, 4G and 5G; Wireless Technologies – WPAN, WLAN, WMAN; Wireless Medium Access Control (MAC) – Overview, SDMA, TDMA, FDMA, CDMA;
2G Technologies: GSM – Services, Architecture, Radio Interface, Protocol Stack, Localization, Call Handling, Handover, Security; 2.5G – GPRS and 3G – EDGE.

Networking: Communication Modes, Basic Network Designs, Cellular Infrastructures

Unit II

Mobile Network Layer and Transport Layer **12 Hours**

Mobile Network Layer: Mobile IP Overview, IP Packet delivery, Agent discovery and advertisement, Registration, Tunneling and Encapsulation, Optimizations, Security, and Dynamic Host Configuration Protocol (DHCP).
Mobile Transport Layer – Motivation, Traditional TCP, Classical TCP Approaches: Indirect TCP, Snooping TCP, Mobile TCP, Transaction-oriented TCP; Optimizations, TCP for 2.5G/3G.

Multi Task gadget: wide area mobile data - air link standards for data - wireless application environment

Unit III

Mobile Ad-hoc Network (MANET) **12 Hours**

Introduction to Mobile ad-hoc networks, Characteristics and features, Applications, Limitations; Routing protocols – Design Issues, Routing algorithms: Proactive (DSDV & OLSR) and Reactive (DSR & AODV), Security in ad hoc networks; Wireless LAN – IEEE 802.11 – System Architecture, Protocol Layers.
Wireless Sensor Network (WSN): Introduction, Architecture, Applications, Properties and Security.

Satellite systems: history - applications - basics - broadcast systems

Unit IV

Mobile Platforms and Applications **12 Hours**

Mobile OS: Overview on Mobile Device Operation Systems (Android, iOS, Black Berry) – Architecture, App

development Kit; Introduction to Network Simulators: Characteristics, Applications, Limitations, Types of Simulators: Wireless Application Protocol (WAP): Introduction, Architecture, Applications.
Application layer Protocols – FTP, SMTP, HTTP, DNS; Windows 10.

Total: 48 Hours

Textbook (s)

1. Raj Kamal, Mobile Computing, Oxford press, Third Edition, 2018
2. Jochen Schiller, Mobile Communications, Pearson Eduction, Secon Edition, 2020

Reference (s)

1. Asoke K Talukder, Hasan Ahmad and RoopaYavagal, Mobile Computing, Second Edition, McGraw Hill, 2010
2. Prasant Kumar Pattnail and Rajib Mall, Fundamentals of Mobile Computing, Second Edition, PHI Learning Pvt. Ltd., 2015
3. Frank Adelstein, et al., Fundamentals of Mobile and Pervasive Computing, McGraw Hill, 2005
4. http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf (NS2 manual)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	30	25	10
Understand	25	15	20
Apply	25	20	30
Analyze	20	20	30
Evaluate	--	10	10
Create	--	10	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. Define Mobile Computing
2. List the applications of mobile computing
3. List the limitations of mobile computing
4. Define Mobile Adhoc Network
5. List the MANET routing issues

Understand

1. Explain mobile computing with architecture
2. Explain TDMA and FDMA with neat diagrams
3. Illustrate the different services provided by the GSM, Explain with system architecture
4. Explain IP Packet delivery with diagram
5. Describe DSSM (Direct Sequence Spread Spectrum)

Apply

1. Illustrate Registration process achieved in mobile network layer
2. Discuss the feature of tunneling and encapsulation
3. Illustrate the function of Dynamic Host Configuration Protocol (DHCP)
4. Discuss about security in Ad-hoc network
5. Illustrate the properties of MANETS

Analyze

1. Compare Snooping TCP and Indirect TCP

2. Analyze the performance of Dynamic Source Routing protocol
3. Compare about iOS and Blackberry Operating Systems
4. Compare NS2 and NS3 simulators
5. Explain about Android Operating System with architecture

Evaluate

1. Evaluate the performance of Wireless Datagram protocol
2. Evaluate the performance of DSDV
3. Evaluate the performance of DSR
4. How Traditional TCP is different from I-TCP
5. How localization and Call handling is performed

Open Book Exam Questions

Q1. Describe the functions of the MS and SIM. Why does GSM separate the MS and SIM? How and where is user-related data represented/stored in the GSM system? How is user data protected from unauthorized access, especially over the air interface? How could the position of an MS (not only the current BTS) be localized? Think of the MS reports regarding signal quality.

Q2. Describe the Concept of GPRS. Using the best delay class in GPRS and a data rate of 115.2 kbit/s – how many bytes are in transit before a first acknowledgement from the receiver could reach the sender (neglect further delays in the fixed network and receiver system)? Now think of typical web transfer with 10 Kbyte average transmission size – how would a standard TCP behave on top of GPRS (see chapters 9 and 10)? Think of congestion avoidance and its relation to the round-trip time. What changes are needed?

21CS006 Distributed Operating Systems

3 0 0 3

Course Outcomes

1. Summarize the fundamental concepts of Distributed Operating Systems.
2. Illustrate the Concepts of Message passing system and Remote Procedure Calls.
3. Design and Implementation of Distributed Shared Memory and Structures.
4. Utilize the Synchronization and Distributed File Systems.
5. Make use of Resource Management and Process Management Concepts.
6. Outline the concept of Naming and Security in Distributed Operating Systems.

CO–PO Mapping

CO	PO1	PO2
1	3	2
2	3	2
3	3	2
4	2	2
5	3	2
6	3	1

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I

Introduction and Communication in Distributed System **12 Hours**

Introduction: Design Issues - Distributed Computing Environment - Message Passing - Features of Good Message Passing System-Issues in IPC by Message Passing-Synchronization- Buffering

Remote Procedure Calls: The RPC Model - Transparency of RPC- Implementing RPC Mechanism - Stub Generation - RPC Messages - Marshaling Arguments and Results - Server Management - Parameter-Passing Semantics- Call Semantics

Communication Protocols for RPCs - Complicated RPCs - Client-Server Binding-Case Study: Sun RPC

Unit II

Distributed Shared Memory and Synchronization **12 Hours**

Distributed Shared Memory: General Architecture of DSM Systems- Design and Implementation Issues of DSM – Granularity - Structure of Shared Memory Space

Synchronization: Clock Synchronization - Event Ordering - Mutual Exclusion –Deadlocks- Election Algorithms

Consistency Models-Replacement Strategy - Thrashing

Unit III

Resource Management and Distributed File Systems **11 Hours**

Resource Management: Features of a Good Global Scheduling Algorithm - Task Assignment Approach – Load Balancing Approach – Load Sharing Approach

Distributed File Systems: Introduction -Desirable Features of a Good Distributed File System - File Models – File Accessing Models – File Sharing Semantics – File Caching Schemes - File Replication

Process Management - Process Migration- Threads- Case Study: DCE Distributed File Service

Unit IV

Naming and Security **12 Hours**

Naming: Desirable Features of a Good Naming System - Fundamental Terminologies and Concepts - System-Oriented Names - Object-Locating Mechanisms - Human-Oriented Names - Name Caches

Security: Potential Attacks to Computer Systems - Cryptography – Authentication

Access Control - Digital Signatures - Design Principles- Case Study: DCE Directory Service

Total: 48 Hours

Textbook (s)

1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2009
2. Andrew S Tannebaum, "Distributed Operating Systems", Pearson Education, 2007

Reference (s)

1. George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design", Fifth Edition, Pearson Education, 2012
2. Tanenbaum A.S., Van Steen M., "Distributed Systems: Principles and Paradigms", Pearson Education, 2007
3. Liu M.L., "Distributed Computing, Principles and Applications", Pearson Education, 2004.
4. Nancy A Lynch, "Distributed Algorithms", Morgan Kaufman Publishers, USA, 2003

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	40	40	--
Understand	30	40	--
Apply	30	20	50
Analyze	--	--	50
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. List any four advantages of Distributed operating systems
2. Define distributed operating systems.
3. List any five reasons why to build distributed operating System.
4. What are the features of good distributed file systems?
5. List the features of a Good Naming System

Understand

1. Explain the design issues of distributed operating systems.
2. Explain the role of Naming in distributed operating systems
3. Summarize the internal and external synchronization of Physical clocks
4. Differentiate Resource Management and Process Management
5. Explain in detail about security and authentication.

Apply

1. Construct the different protocols of RPC.
2. Identify the File shearing Semantics and file caching Schemes.
3. Develop the Bully and Ring Algorithm
4. Develop the scenario how might the clocks in two computers that are linked by a local network be synchronized without reference to an external time source
5. Utilize the various deadlock prevention techniques with suitable example?

Analyze

1. Analyze the Concept of exception handling in RPC.
2. Examine the Concept of Thread.
3. Compare Fault tolerance and Thrashing.
4. Classify the deadlock avoidance techniques.
5. Simplify an algorithm using multicast and logical clocks for mutual exclusion.

Open Book Exam Questions

Question 1:

A distributed system has DSM facility. The process-scheduling mechanism of this system selects another process to run when a fault occurs for the currently running process, and theCPU is utilized while the block is being fetched. Two system engineers arguing about how to better utilize the CPUs of this system have the following opinions:

(a) The first one says that if a large number of processes are scheduled for execution at anode, the available memory space of the node can be distributed among these processes that almost always there will be a ready process to run when a page fault occurs. Thus, CPU utilization can be kept high.

(b) The second one says that if only a few processes are scheduled for execution at a node, the available memory space of the node can be allocated to each of the few processes, and each process will produce fewer page faults. Thus, CPU utilization can be kept high. Whose argument is correct? Give reasons for your answer.

Question 2:

The password mechanism is used in a distributed system to authenticate users at login time. State the most suitable locations (according to you) for storing the login program and the password file in the following cases:

(a) The distributed system is based on the workstation-server model with each workstation having a small hard disk of about 20 megabytes capacity.

(b) The distributed system is based on the workstation-server model. Some of the workstations are diskless and others have a small hard disk of about 20 megabytes capacity.

(c) The distributed system is based on the processor-pool model.

21DS507 Deep Learning Lab

0 0 3 1.5

Course Outcomes

At the end of the course, the students will be able to:

1. Design and implement the basic structure of neural network.
2. Implement the forward and backward propagation.
3. Simulate various learning techniques in neural network.
4. Learn and process elements in the training set using CNN algorithm.
5. Implement transfer learning algorithm.
6. Study of recurrent neural network

CO–PO Mapping

CO	PO4	PO5	PO8
1	3	2	2
2	3	3	2
3	3	3	2
4	2	2	3
5	2	2	2
6	1	1	3

3–Strongly linked **|** 2–Moderately linked **|** 1–Weakly linked

Lab Manual

1. Deep Learning for Data science Laboratory– AIDS Department, GMRIT

Textbook (s)

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron, October 2022: Third Edition.
2. Fundamentals of Deep Learning: Designing Next-generation Machine Intelligence Algorithms by Nicholas Locascio and Nikhil Buduma O'Reilly Media; 1 edition (June 29, 2017)
3. Simon S. Haykin, Neural Networks, Prentice Hall, 2nd edition
4. B. Yegnanrayana , “Artificial Neural Networks” , PHI.

Reference (s)

1. Franchois Chollet, Deep Learning with Python
2. Deep Learning: A Practitioner's Approach by Adam Gibson and Josh Patterson Shroff/O'Reilly; First edition (2017)
3. Python Deep Learning by Daniel Slater and Gianmario Spacagna, Packt Publishing; 2/e (January 16, 2019)
4. Bishop, C., M., Pattern Recognition and Machine Learning, Springer, 2006
5. Kevin P. Murphy., Machine Learning: A Probabilistic Perspective

List of Experiments

Week 1: Define the Structure and Parameter’s Initialization in a Neural Network Model.

Week 2: Implement the Forward propagation and Compute the Cost Function.

Week 3: Implement Back propagation to get the gradients and Update parameters (gradient descent)

Week 4: Build a Neural Network Model and predict the Output.

Week 5: Implement the Hebbian Learning neural networks.

Week 6: Digit and Character recognizer using CNN.

Week 7: Explore implementation of AlexNet.

Week 8: Exploring Different CNN Models: VGG16, VGG19

Week 9: Exploring Different CNN Models: GoogleNet,

Week 10: Explore implementation of VGG16 with transfer learning and fine tuning

Week 11: One-hot encoding of words and characters using word embedding.

Week 12: Study the construction and working of Recurrent Neural Network.

Week 13: Explore LSTM

Week 14: GRU to predict stock prices based on historic data.

Week 15: Explore applications like Image segmentation and object detection.

Week 16: Automatic image captioning and video to text models.

Augmented Experiments:

(Students shall opt any one of the Augmented Experiments in addition to the regular experiments)

a. Image Segmentation
b. Object Detection
c. Human Facial Recognition.
d. Automatic image captioning
e. Stock Market Prediction.
f. Social Media
g. Aerospace
h. Defense
i. Healthcare
j. Signature Verification.
k. Social media
l. Aerospace
m. Defense
n. Healthcare
o. Handwriting Analysis
p. Speech Recognition
q. Weather forecasting

Open Elective-II

21DS001 Fundamentals of Data Science

Course Outcomes

1. Understand the key steps and pipeline of Data Science and its application in solving real-world problems.
2. Recognize the importance of measuring similarity and dissimilarity between features in data for various analysis tasks.
3. Appreciate the significance of pre-processing techniques in preparing data for analysis in real-time scenarios.
4. Identify the characteristics and practical applications of different regression models used in real-world scenarios.
5. Evaluate classification models using appropriate metrics, including the confusion matrix, to assess model performance and make informed decisions.
6. Understand the principles of ensemble modeling and clustering, and apply appropriate ensemble techniques to improve the accuracy and reliability of machine learning models.

COs–POs Mapping

COs	PO3	PSO1	PSO2
1	2	2	2
2	2	3	3
3	2	3	3
4	2	2	2
5	2	2	2
6	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

UNIT-I 12+8 hours

Introduction: Relation among AI, ML and Data Science, Importance of Data Science; Data Science Process; **Data Exploration:** Objectives of Data Exploration, Forms of Data(Structured, Semi Structured, Unstructured),Datasets(data objects and types of attributes/fields), Characteristics of Datasets and corresponding Statistical Measures;

Data Visualization: Univariate Visualization, Multivariate Visualization.

Categorization of Data Science Algorithms. Overview of different kind of dataset (i.e. text, image) and the different format(ie. CSV, json).

Practical Components

1. Perform data exploration and statistical analysis on a real-world dataset to understand its characteristics and calculate relevant statistical measures.
2. Explore datasets of different formats (text and image) and apply data exploration techniquesaccordingly.
3. Visualize a dataset using univariate and multivariate visualization techniques to analyze the distribution of data and explore relationships between variables.
4. Analyze a dataset using categorical data visualization techniques (bar plots, pie charts) to gain insights into demographic patterns and distributions.

UNIT-II 12+8 hours

Data Similarity/Dissimilarity: Understanding data similarity and dissimilarity, Measures for comparing different types of data (nominal, ordinal, binary, numerical).

Data Preprocessing: Data Preprocessing Pipeline, Preprocessing techniques for cleaning and integrating data, Data reduction techniques for handling large datasets, Cosine Similarity.

Distance based similarity (Euclidean distance, Jaccard Similarity).

Practical Components

1. Apply data preprocessing techniques and impute missing values on a real-world dataset.
2. Select a large dataset and apply data reduction techniques such as feature selection and

dimensionality reduction (e.g., PCA, t-SNE) to handle its size while preserving important information and patterns in the data.

3. Calculate the cosine similarity between pairs of text documents to measure their similarity.
4. Integrate multiple datasets and perform data cleaning to create a unified and consistent dataset.

UNIT-III 12+8 hours

Regression: Introduction to linear regression for forecasting numerical quantities, Logistic regression for classification problems, Regularization techniques for improving model performance;

Classification: ClassificationPrinciples,ClassificationModelEvaluationMetrics(ConfusionMatrix), Classification using Decision Trees, Distance based Classifier(k-NN),Bayesian classifier.

Regression vs Classification.

Practical Components

1. Perform linear regression on a dataset to forecast a specific target variable. Evaluate theperformance of the regression model using appropriate evaluation metrics.
2. Apply the Decision Trees algorithm to build a classification model. Utilize appropriate evaluationmetrics and construct a confusion matrix to assess the model's performance.
3. Perform linear regression on a dataset and apply regularization techniques such as L1 or L2regularization to improve the model's performance.
4. Implement a Bayesian classifier to classify emails as spam or non-spam. Use appropriate evaluation metrics to assess the classifier's performance.

UNIT-IV 12+8 hours

Ensemble Learning: Conditions for Ensemble Modeling, Overview of ensemble techniques(Voting, Bagging,Boosting and Random Forest);

Clustering: Clustering Principles, Clustering for description/preprocessing/classification, Types of Clustering, Clustering Evaluation Parameters, Clustering Algorithms (k-Means) and Evaluation metrics for assessing the quality of clustering results;

Applications/ Purpose of Clustering.

Practical Components

1. Explore ensemble learning techniques such as Random Forest on a dataset suitable for classification or regression.
2. Apply the K-means clustering algorithm on a dataset for classification purposes. Use evaluation metrics to assess the quality of the clustering results and experiment with different values of k.
3. Use a voting classifier to combine multiple classification algorithms and improve the accuracy of thefinal model.
4. Perform Hyperparameter tuning for any Ensemble Algorithms(ie. RF)

Total: 48+32 Hours

Textbooks:

1. Vijay Kotu & Bala Deshpande, “Data Science: Concepts and Practice”, 2ndEdition, MorganKaufmann Publication, 2017.
2. JlaweiHan&MichelineKamber,“DataMining:ConceptsandTechniques”,3rdEdition,MorganKau f mannPublishers,2012.
3. Pang-NingTanetal.,“IntroductiontoDataMining”,2ndEdition,PearsonPublications,2019.

References:

1. Vijay Kotu & Bala Deshpande, “Data Science: Concepts and Practice”, 2ndEdition, MorganKaufmann Publication, 2017.
2. TomM.Mitchell,"MachineLearning",TataMcGrawHill,1997.
3. SunilaGollapudi,“PracticalMachineLearning”FirstEdition2016,PacktPublishingLtd.

Internal Assessment Pattern

Cognitive Level	**Int.Test1(%)**	**Int.Test1(%)**	**OpenbookTest1(%)**
Remember	20	20	--
Understand	50	30	--
Apply	30	30	80
Analyze	--	20	20
Evaluate	--	--	--
Create	--	--	--
Total(%)	**100**	**100**	**100**

SAMPLEQUESTION(S)

Remember

1. What are the different forms of data in data science? Provide examples for each form.
2. Name the statistical measures used to analyze datasets in data science.

Understand

1. Why is pre-processing important in machine learning? Explain with relevant examples
2. What is the difference between supervised and unsupervised algorithms? Provide an example for each type.

Apply

1. Imagine you have a dataset containing information about houses. How would you handle the followingscenarios using pre-processing techniques:

   a. Data Cleaning: How would you deal with missing values in the dataset? What approach wouldyou take to handle outliers or inconsistent data?

   b. Data Integration: Suppose you have additional datasets with relevant information about houses,such as neighborhood demographics or school ratings. How would you combine this information with the existing dataset?

   c. Data Reduction: The dataset contains numerous features. How would you simplify the datasetby reducing the number of features without losing important information?

   d. Data Transformation and Discretization: How would you standardize numerical features in thedataset? Additionally, how would you convert continuous variables, such as house prices, into categorical variables for analysis purposes?

Open Book Question(s)

1. Discuss the importance of data visualization in data science. Provide examples of univariate and multivariate visualizations, and explain how they can be used to gain insights from complex datasets. Support your answer with relevant illustrations or diagrams.
2. Compare and contrast supervised and unsupervised learning algorithms. Provide a detailed explanation of each type, including their characteristics, applications, and differences in the training process. Give an example scenario where you would use each type of algorithm.
3. Assume you have been given a dataset with various features related to customer demographics and their purchasing behavior. Design a data science pipeline to address the following tasks:
   a. Data exploration and statistical analysis of the dataset.
   b. Preprocessing steps, including handling missing values, data cleaning, and feature transformation.
   c. Applying a suitable regression model to predict customer spending based on the available features.
   d. Evaluating the performance of the regression model using appropriate evaluation metrics.

21TPX01 Term Paper

0 0 3 1.5

Course Outcomes

1. Interpret the literature to link the earlier research with the contemporary technologies
2. Communicate effectively as an individual to present ideas clearly and coherently
3. Review the research findings and its correlation to the latest applications
4. Prepare documents and present the concepts clearly and coherently
5. Inculcate the spirit of enquiry for self-learning
6. Identify interdisciplinary oriented topics

COs – POs Mapping

COs	PO1	PO4	PO10	PO12
1	-	2	-	-
2	-	-	3	3
3	3	-	-	-
4	-	-	3	-
5	-	-	-	3
6	1	-	-	-

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Term Paper: The term paper is a self-study report with an objective enhancing the students' written technical-communication and shall be carried out in 5th semester. Every student will take up this either individually or as a batch of maximum four members and submit a report. The scope of the term paper could be an exhaustive literature review choosing any engineering concept with reference to standard research papers or an extension of the concept of earlier course work in consultation with the term paper supervisor.

The report will be evaluated by a committee nominated by HoD with the approval of the Controller of Examinations (CoE)

21ESX02 Employability Skills II

0 0 2 0

Course Outcomes

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Introspect & develop life skills with constructive approach
3. Assess and improve analytical skills
4. Understand the GUI of WEKA tool and can create new ARFF data.
5. Discover different Pre-Processing Techniques for different kinds of databases
6. Use Association rule mining algorithms to generate association rules

COs – POs Mapping

COs	PO1	PO2	PO3	PO5	PO 6	PO8	PO10	PO12
CO1	-	-	-	-	-	-	3	-
CO2	-	-	-	-	3	-	2	2
CO3	2	-	-	-	-	2	-	-
CO4	3	-	3	2	-	2	-	-
CO5	3	2	3	2	-	-	-	2
CO6	2	2	3	2	3	-	-	-

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Soft Skills:

Sl No.	5th Semester (Topic & Content)	No. of Periods
1.	**Introduction to Campus Placements:** Stages of Campus Placement, Skills assessed in Campus Placements & How to get ready?	02
2.	**Motivational Talk on Positive Thinking:** Beliefs, Thoughts, Actions, Habits & Results (Success)	02
3.	**Resume Preparation:** Resume? Templates? Mistakes to be avoided in a Resume, Steps to be followed in preparing it.(with examples)	03
4.	**Group Discussions** (Recap)**:** GD? Stages of a GD, Skills assessed in a GD, Blunders to be avoided, How to excel in a GD? (through Practice Sessions)	03
5.	**Psychometric Tests:** Definition, Types of Psychometric Tests: Numerical Computation, Data Interpretation, Verbal Comprehension, Verbal Critical Reasoning and Personality Questionnaires	03
6.	**Exercises related to Communication:** Story Writing, TAT etc	03
	Total Periods	**16**

Quantitative Aptitude:

Sl No.	Semester-V (Topic)	No. of Periods
1.	Square &Cube roots	02
2.	Partnership	02
3.	Partnership	02
4.	Logarithms	02
5.	Progressions	02
6.	Mensuration	02
7.	Mensuration	02
8.	Data Sufficiency	02
	Total Periods	**16**

Module-1:

No of Hours: 16 Hours

Introduction: Importance of Competitive Programming, Conditional Statements, Loops, Nested Loops. Strings, Arrays, Recursion and pointers

Practice Problems:

Week	Topic	Practice Problem Links
1	Introduction to CP	Creating accounts on Global Coding Platforms
2	Conditional Statements	https://www.codechef.com/practice/tags/conditional-statements
3	Loops	https://www.codechef.com/practice/tags/loops
		https://leetcode.com/tag/math/
		https://www.interviewbit.com/practice/#topics[]=math
4	Nested Loops	https://www.codechef.com/practice/topics/math
		https://www.interviewbit.com/practice/#topics[]=math
5	Strings	https://www.codechef.com/practice/topics/strings
		https://leetcode.com/tag/string/
		https://www.interviewbit.com/practice/#topics[]=strings
6	Arrays	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=arrays
		https://leetcode.com/tag/array/
		https://www.codechef.com/practice/topics/arrays
		https://www.interviewbit.com/practice/#topics[]=arrays
7	Two Pointer	https://www.interviewbit.com/practice/#topics[]=two-pointers
		https://leetcode.com/tag/two-pointers/
8	Recursion	https://leetcode.com/tag/recursion/
		https://www.codechef.com/practice/tags/recursion

Module-2:

No of Hours: 16 Hours

Searching, Sorting, Linked List, Linked List: Double Linked List, Circular Linked List, Stack, Queue, hashing

Practice Problems:

Week	Topic	Practice Problem Links
9	Searching-I	https://www.codechef.com/practice/topics/binary-search
		https://leetcode.com/tag/binary-search/
		https://www.interviewbit.com/practice/#topics[]=binary-search
		https://www.codechef.com/practice/tags/searching
10	Searching-II	https://www.codechef.com/practice/topics/binary-search
		https://leetcode.com/tag/binary-search/
		https://www.interviewbit.com/practice/#topics[]=binary-search
		https://www.codechef.com/practice/tags/searching
11	Sorting-I	https://www.codechef.com/practice/topics/sorting
		https://leetcode.com/tag/sorting/
		https://leetcode.com/tag/merge-sort/
12	Sorting-II	https://www.codechef.com/practice/topics/sorting
		https://leetcode.com/tag/sorting/
13	LinkedList: Double LinkedList, Circular LinkedList	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=linked-lists
		https://leetcode.com/tag/linked-list/
		https://www.interviewbit.com/practice/#topics[]=linked-lists
		https://leetcode.com/tag/doubly-linked-list/
14	Stack	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=stacks
		https://leetcode.com/tag/stack/
		https://www.interviewbit.com/practice/#topics[]=stacks-and-queues
15	Queue	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=queues
		https://leetcode.com/tag/queue/
		https://www.interviewbit.com/practice/#topics[]=stacks-and-queues
16	Hashing	https://www.interviewbit.com/practice/#topics[]=hashing

Total: 32 Hours

References:

1. https://leetcode.com
2. https://www.codechef.com
3. https://www.hackerearth.com
4. https://www.interviewbit.com
5. https://www.hackerrank.com

21HSX12 CC & EC Activities II

0 0 1 0

Course Outcomes

1. Interpret and present the abstractive technical information through an activity
2. Think critically in providing solutions to the generic and common problems
3. Demonstrate the creative thinking in dealing with liberal arts
4. Instill team sprit through active engagement with the peer
5. Develop programs of common interest having social impact
6. Empower the under privileged through motivational activities

COs – POs Mapping

COs	PO6	PO7	PO9	PO10
1	-	-	-	3
2	3	2	-	-
3	3	-	-	-
4	-	-	3	-
5	3	-	-	-
6	3		-	-

Co-Curricular and Extra Curricular (CCEC) Activities:

Students shall acquire 1 credit each in 2nd and 3rd years with the following scheme:

Scheme of evaluation for the CCEC activities:

- No. of slots in each Semester @ 2 slots every week : 24
- No. of Stream (2-CC + 1-EC) : 3
- No. of slots allotted for each stream : 8

Requirement for the award of 1- Credit

- Students shall choose at least two streams of events in each semester
- Students shall secure 75% attendance in each stream of events to obtain a certificate
- Students shall obtain 2 certificates of Participation in each semester.

The credits earned through these courses will be indicated in the grade sheet and will not beaken into account for CGPA calculation.

21SIX01 Summer Internship I

0 0 0 1

Course Outcomes

1. Demonstrate the application of knowledge and skill sets acquired from the course and workplace in the assigned job function/s
2. Solve real life challenges in the workplace by analyzing work environment and conditions, and selecting appropriate skill sets acquired from the course
3. Articulate career options by considering opportunities in company, sector, industry, professional and educational advancement
4. Communicate and collaborate effectively and appropriately with different professionals in the work environment through written and oral means
5. Demonstrate the ability to harness resources by examining challenges and considering opportunities
6. Demonstrate appreciation and respect for diverse groups of professionals by engaging harmoniously with different company stakeholders

COs – POs Mapping

COs	PO1	PO2	PO8	PO10	PO12
1	3	-	-	-	-
2	3	-	-	-	-
3	-	-	-	-	3
4	-	-	-	3	-
5	-	2	-	-	-
6	-	-	3	-	-

Summer Internship: As a part of curriculum in all branches of Engineering, it is mandatory for all students to undergo summer internship Programme at industries (core or allied) / R & D organization to get practical insight of their subject domain during summer break after the 4th semester. This summer internship Programme shall be availed to a maximum duration of 4 weeks and the assessment shall be carried out with both internal and external experts leading to “Satisfactory” and “Non-Satisfactory Performance”, and it will not be accounted for the calculation of CGPA.

Audit Course

0 0 0 0

Course Outcomes

1. Interpret the meaning of values and select their goals by self- Investigation based on personal values activity
2. Interpret the major events and issues related to a period in Indian history
3. Assess the benefits and limitations of science and its application in technological developments towards human welfare
4. Check the awareness regarding basic human rights and to uphold the dignity of every individual
5. Assess the individual and group behaviour, and understand the implications of organizational behaviour on the process of management
6. Determine the appropriateness of various leadership styles and conflict management strategies used in organizations

COs – POs Mapping

COs	PO12
1	3
2	3
3	3
4	2
5	3
6	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Audit Courses: Audit courses are among the compulsory courses and do not carry any credits. All the students shall register for one Audit courses in the beginning of 3rdsemester. List of the courses will be notified at the beginning of the third semester for all students and the student has to choose one audit course under self-study mode at the beginning of third semester. All the students (regular and lateral entry students) shall complete the audit course similar to other regular courses and the results will be indicated with “Satisfactory” or “Not Satisfactory” performance.

6th Semester

21DS601 Optimization techniques for machine learning

3 0 0 3

Course Outcomes

1. Explain the Data Analysis Fundamentals
2. Illustrate various data Visual aids
3. Categorize different data transformation and descriptive statistics
4. Identify different Correlation and Inferences from statistical tests
5. Make use of the concept Hypothesis Testing
6. Classify different Multivariate Analysis techniques

CO–PO Mapping

CO	PO2	PO3	PSO1	PSO2
1	1	2	2	2
2	1	2	2	2
3	2	2	2	2
4	2	2	2	2
5	2	2	2	2
6	2	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12 Hours**

Introduction to Optimization: Introduction to optimization problems and applications in machine learning, Convexity, convex functions, and convex optimization

Unconstrained Optimization: Gradient descent, Newton's method, and their variants, stochastic gradient descent

Proximal Methods, Online Learning, Distributed and Parallel Optimization

Unit II **12 Hours**

Constrained Optimization: Linear programming and its applications in machine learning, Quadratic programming and its applications in machine learning, Nonlinear programming and its applications in machine learning

Convex Optimization: Convex optimization algorithms: sub gradient methods, projected gradient methods, interior point methods. Duality theory and its applications in machine learning Convex relaxation and its applications in machine learning

Karush-Kuhn-Tucker (KKT) Conditions, Duality in Convex Optimization, Robust Optimization

Unit III **12 Hours**

Large Scale Optimization: Mini-batch gradient descent and stochastic gradient descent, Distributed optimization algorithms, Optimization for online learning

Regularization and Sparsity: L1 and L2 regularization, Elastic net regularization and its applications in machine learning, Group sparsity and its applications in machine learning

Optimization on Parallel Architectures, Sparse Recovery and Compressed Sensing

Unit IV **12 Hours**

Advanced Optimization Techniques: Conjugate gradient methods and their variants, Quasi-Newton methods and their variants, Trust region methods and their variants

Optimization for Deep Learning: Back propagation and optimization in deep learning, Optimization for recurrent neural networks and long short-term memory networks, Optimization for generative models, such as variational auto-encoders and generative adversarial networks

Gradient Clipping, Weight Initialization Techniques, Adaptive Learning Rate Methods

Total: 48 Hours

Textbook (s)

1. "Convex Optimization" by Stephen Boyd and Lieven Vandenberghe
2. "Optimization for Machine Learning" by Suvrit Sra, Sebastian Nowozin, and Stephen Wright
3. "Numerical Optimization" by Jorge Nocedal and Stephen J. Wright

Reference (s)

1. "Stochastic Gradient Descent and Beyond" by Martin Jaggi
2. "Sparse Modeling: Theory, Algorithms, and Applications" by Irina Rish and Genady Grabarnik

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	50	40	--
Understand	30	40	--
Apply	20	20	50
Analyze	--	--	50
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	100

SAMPLE QUESTION (S)

Remember

1. What is optimization
2. Define purpose of optimization

Understand

1. Explain different convex optimization
2. Explain the advanced optimization techniques

Apply

1. Apply various visual aids to identify behavior of data
2. Make use of Multivariate Analysis methods for data analysis

Analyze

1. Compare and contrast various L1 and L2 regularization

Open Book Exam Questions

Question 1:
Perform below analysis on a house price prediction dataset.

a) Univariate Analysis
 iii. Analysis of a numerical feature
 iv. Analysis of a categorical feature
b) Bivariate Analysis
 v. Relationship of a numerical feature with another numerical feature
 vi. Relationship of a numerical feature with a categorical feature

Question 2:
Perform below analysis on weather prediction dataset.

1. Correlation Analysis
 i. Correlation Heat Map
 ii. Zoomed Heat Map
2. Investigation of missing values
 i. What's missing? to what extent?
 ii. Visualizing missing values in a data frame

21IT602 Automata and Compiler Design

3 0 0 3

Course Outcomes:

At the end of the course students are able to:

1. Analyses and design finite automata, pushdown automata for the formal languages and grammars.
2. Apply working skills in theory and application of finite state machines in compiler design.
3. Demonstrate a compiler that includes all the phases of Compiler Design.
4. Show recursive descent-production rules-parsing-and language semantics in high level compilers.
5. Demonstrate the machine dependent and independent codes.
6. Judge the ideas-the techniques-and the knowledge acquired for the purpose of other software design.

CO – PO Mapping

COs	**PO$_1$**	**PO$_2$**	**PO$_3$**
1	1	3	2
2	2	3	3
3	3	2	2
4	2	1	3
5	2	1	3
6	2	1	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Syllabus

Unit I **13 Hours**

Finite Automata & Lexical Analysis

DFA - NFA - Recognition of a language by an Automaton - Equivalence of DFA and NFA - Finite Automata with Null-Closure - Minimization of FA - Equivalence of FAs, Regular Sets and Languages - Equivalence of FA & regular expression Overview of Language Processing-Compiler-Assembler-Interpreters-Linkers & Loaders-Structure of Compiler-Phases of a Compiler-Lexical Analysis-Role of Lexical Analysis-Token-Patterns and Lexemes-Lexical Errors-

Finite Automata with output: Mealy and Moore Machines

Unit II **12 Hours**

Push-Down Automata & Syntax Analysis

Non-regular Languages - CFLs - Closure Properties of CFLs - CFGs - derivation trees - Simplification - Ambiguity Push-Down Automata - Normal Forms - Chomsky Hierarchy ,Top-down Parsing-First and Follow-LL(1) Grammar-Non-Recursive Predictive Parsing

Handling Ambiguous grammars-Error recovery in LR parsing

Unit III **12 Hours**

Bottom-up parsing-Shift Reduce Parsing

Model of an LR Parsers-Construction of SLR Tables-Construction of CLR (1)- LALR Parsing tables-Dangling ELSE ambiguity. Semantic Analysis-SDT-Intermediate Code -Three Address Code-Quadruples-Triples-Indirect Triples-Abstract SyntaxTrees-DAG for Expressions.

Data structures used in symbol table

Unit IV **11 Hours**

Optimization

Machine Independent Code Optimization-Common Sub-expression Elimination-Constant Folding-Copy Propagation-Dead Code Elimination-Strength Reduction-Loop Optimization-Basic Blocks-Flow Graph-DAG for basic Blocks-Machine Dependent Code Optimization: Peephole Optimization-Register Allocation-Instruction Scheduling

Code generation algorithm-optimization among basic blocks

Total: 48 Hours

Textbook (s)

1. J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, 3rd Edition,Pearson/Addison Wesley, 2007
2. Mishra & Chandra Sekharan, Theory of Computer Science& Automata Language and Computation, 3rd Edition,Prentice Hall of India, 2007.
3. Alfred V Aho, Monical S Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers, Principles Techniques and Tools, 2ndEdition, Pearson, 2007.
4. V. Raghavan, Principles of Compiler Design, 2nd Edition, TMH, 2011.

Reference (s)

1. P. Linz, Introduction to Formal Language and Computation, 2nd Edition, Narosa, 2006.
2. H. R. Lewis & C. H. Papadimitriou, Elements of the Theory of Computation, Prentice Hall of India, 2nd Edition -2006.
3. Nandini Prasad, Principles of Compiler Design, 2nd Edition, Elsevier, 2012
4. Kenneth C Louden, Compiler Construction, Principles and Practice, 1st Edition, Cengage, 1997.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	20	20	--
Understand	50	40	--
Apply	30	40	70
Analyze	--	--	20
Evaluate	--	--	10
Create	--	--	--
Total (%)	100	100	100

Sample Question (S)

Remember

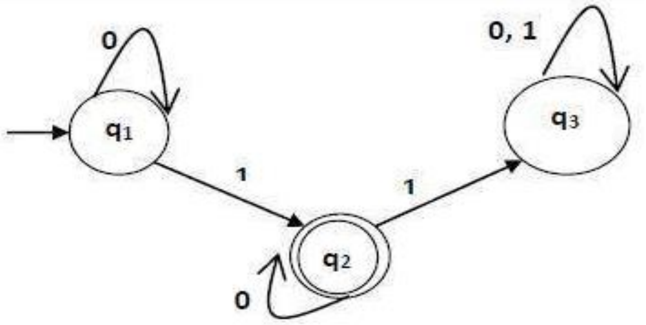
1. List out the different tuples of Deterministic Finite Automata
2. List out the closure properties of regular sets
3. State some software tools that manipulate source program.
4. List out the cousins of compiler.

Understand

1. Explain the procedure of simplification of CFG
2. Describe the types of Turing machines
3. Illustrate the pumping lemma with an example
4. Represent the Error-recovery actions in a lexical analyzer.
5. Formulate the roles and tasks of a lexical analyzer.
6. Illustrate Why lexical and syntax analyzers are separated out.
7. Identify the problems with top down parsing

Apply

1. Apply the Arden's theorem to convert the following Finite automata to Regular expression



2. Apply Pumping lemma to prove the language L={ ap / p is a prime number} is not regular
3. Eliminate UNIT productions from the CFG:

   S→A
   A→B
   B→a/b/c

4. Demonstrate top down parsing and bottom up parsing methods.
5. Show the contents of activation record.
6. Choose the properties of optimizing compiler.

Analyze

1. Compare and Contrast Mealy machine and Moore machine
2. Analyze the solution to the PCP with the two lists M=(abb ,aa, aaa) and N=(bba, aaa, aa)
3. Analyze whether the language L ={anbncn/n>=1} is Context free language or not
4. Identify the difficulties with top down parsing.
5. Breakdown the grammar for flow-of-control statements.
6. Differentiate declarations are done in a procedure using syntax directed translations

Evaluate

1. Assess the performance of Pushdown automata over Finite automata
2. “Regular sets are closed under union” is true or false? Justify your answer
3. “We can design PDA for L ={anbncn/n>=1}” is true or false? Justify your answer
4. Determine the addressing mode and associated costs in the target machine.
5. Defend the step to partition a sequence of 3 address statements into basic blocks.
6. Choose the important classes of local transformations on basic blocks.
7. Criticize code optimization and optimizing compiler.

Open Book Exam Questions

Give regular expressions for the following sets of strings. You may only use basic regular expressions formed from characters and epsilon (ε),character classes denoting a single character ([…] and [^…]), concatenation (xy), alternation (x|y), repetition (x* and x+), and optional (x?). You may also give names to subexpressions (name=re) and use parentheses for grouping.

(a) Identifiers formed as follows: an identifier consists of one or more letters (a-z and A-Z), digits (0-9) and underscores (_). An identifier must begin with a letter and may not end with an underscore.

(b) Remote file identifiers of the form user@hostname:filename, constructed as follows. The parts of the identifier are made up of words, which are sequences of one or more letters and digits. The user part contains a single word. A hostname consists of one or more words separated by periods, like www.google.com orattu. A filename consists of one or more words separated by slash (/) characters with an optional leading and/or trailing slash (standard Unix conventions). The user@ part is optional and may be omitted. The entire user@hostname: part may be omitted, including the trailing colon. The user@ part may not appear unless the hostname: part is also included.

1. Draw a DFA that accepts the same set of strings generated by the regular expression p(p|q)*p
2. Convert the following into three address statements.

```
          main()
          {inti, a[5],b[5];
for (i=0;i<5;i++)
a[b[i]] = b[a[i]];
          }
```

3. For this C program segment, construct the tree and generate the assembler code you are allowed to use pseudo assembler code.

```
          for (i=0;i<10;i++)
a[i] = random() % 2;
```

21CS603 Software Engineering

3 0 0 3

Course Outcomes

1. Explain the need of Software Life Cycle Models
2. Build end-user requirements into system and software requirements,
3. Summarize the system models of software engineering
4. Identify and apply appropriate software architectures and patterns to carry out high level design
5. Choose various testing techniques during software development
6. Categorize Risk management and Software quality for software products

COs–POs Mapping

Cos	PO4	PO5	PO8	PO11	PSO1
1	3	3	2	2	2
2	3	3	2	2	2
3	3	2	2	2	2
4	2	2	2	2	2
5	2	3	2	2	2
6	2	3	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12 Hours**

Introduction to Software Engineering and SDLC, Software Myths, CMMI, Process models: Linear Sequential model, Prototyping model, Evolutionary models: Spiral model, Agile developmental methodologies-Scrum & XP

Incremental model, software development : Product based and application based

Unit II **12 Hours**

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification. Software Requirements Engineering Process, Feasibility studies, Requirements elicitation and analysis, requirements validation.

System models: Context models, behavioral models, data models, object models.

Structure of Software Requirements Document, Structured analysis methods

Unit III **12 Hours**

Design concepts, data design, software architecture, Architectural styles and patterns, User interface design - Golden rules, User interface analysis and design and steps. Conceptual model of UML, basic structural modeling, Static and Dynamic UML diagrams : class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, etc.,

Data Acquisition System - Monitoring and Control System

Unit IV **12 Hours**

Testing strategies and Risk Management: Testing levels: Unit testing, integration testing, system testing – alpha and beta testing, Testing Types: black box and white box testing techniques, Cyclomatic Complexity, debugging, Risk management - Risk types, strategies, estimation and Planning. Software Quality - Quality assurance and its techniques

Software measurement, metrics for software quality

Total: 48 Hours

Textbook (s)

1. Roger S. Pressman, Software Engineering, A practitioner's Approach, 8th Edition, McGraw-Hill International Edition, 2015
2. I. Sommerville, Software Engineering, 7thEdition, Pearson education, 2004.
3. Rajib Mal, Fundamentals of software Engineering, 4th Edition, Eastern Economy Edition, 2014.

Reference (s)

1. K K Aggarwal and Yogesh singh, Software engineering,3rd Edition, New age international publication,2008

SAMPLE QUESTION (S)

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)	Open Book Exam (%)
Remember	30	30	--
Understand	40	40	--
Apply	30	30	50
Analyze	--	--	50
Evaluate	--	--	--
Create	--	--	
Total (%)	100	100	100

Remember

1. Define software engineering.
2. List different types software myths.

Understand

1. Applying the process of requirement analysis, discuss how the requirements can be collected for aproject.
2. Applying debugging strategy fin an error from a code?

Apply

1. Applying the process of requirement analysis, discuss how the requirements can be collected for aproject.
2. Applying debugging strategy fin an error from a code?

Analyze

1. Compare and Contrast software life cycle models.
2. Analyze risk types in the risk management.

Open Book Exam Questions

Assume that 10 errors have been introduced in the requirements model and that each error will be amplified by a factor of 2:1 into design and an addition 20 design errors are introduced and then amplified 1.5:1 into code where an additional 30 errors are introduced. Assume further that all units testing will find 30 percent of all errors, integration will find 30 percent of the remaining errors, and validation tests will find 50 percent of the remaining errors. No reviews are conducted. How many errors will be releasedto the field.

21MLC12 Machine Learning for Business Intelligence

3 0 2 4

Course Outcomes

1. Understand the basic concepts of business analytics.
2. Identify the application of business analytics and use tools to analyze business data.
3. To know how to derive meaning form huge volume of data and information and knowledge discovering process is used in business decision making.
4. Become familiar with various metrics, measures used in business analytics.
5. Illustrate various descriptive, predictive, and prescriptive methods and techniques.
6. Model the business data using various business analytical methods and techniques.

CO–PO Mapping

CO	PO2	PO3	PSO1	PSO2
1	2	2	2	2
2	2	2	3	3
3	2	2	2	2
4	2	2	2	2
5	2	2	3	3
6	2	2	3	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

UNIT I **11+ 9 Hours**

Introduction to Business Analytics: Introduction to Business Analytics, need and science of data driven (DD) decision making, Descriptive, predictive, prescriptive analytics and techniques, big data analytics, Web and Social media analytics, Machine Learning algorithms, framework for decision making, challenges in DD decision making and future.

Recommender Systems, Automated Decision-Making, Reinforcement Learning for Decision-Making

Practical Components

1. Descriptive, Predictive analysis on numeric data
2. Implementation of central tendency models on an appropriate dataset
3. Sample case study on Types of Social Media Analytics Report
4. Sample case study on Decision making

UNIT II **12+ 9 Hours**

Data mining, Text mining, Web mining, Spatial mining, Process mining, Data warehouse and datamarts. Data mining process KDD, CRISP-DM, SEMMA and Domain-Specific, Classification and Prediction performance measures -RSME, MAD, MAP, MAPE, Confusion matrix, Receiver Operating Characteristic curve & AUC; Validation Techniques - hold-out, k-fold cross-validation, LOOCV, random subsampling, and bootstrapping.

Nested Cross-Validation, Stratified Cross-Validation, Time Series Cross-Validation

Practical Components

1. Measure Prediction performance with confusion matrix on sample dataset
2. Implement ROC curve and AUC
3. Implementation of K-fold cross validation for sample dataset
4. Implementation of bootstrapping with suitable datasets

UNIT III **11+ 6 Hours**

Forecasting Techniques: Introduction, time-series data and components, forecasting accuracy, moving average method, single exponential smoothing, Holt's method, Holt-Winter model, Croston's forecasting method, regression model for forecasting, Auto regression models, autoregressive moving process, ARIMA, Theil's coefficient.

Coefficient of Variation, Shannon Diversity Index, Simpson's Diversity Index, Gini Coefficient

Practical Components

1. Implementation of various smoothing techniques on sample datasets
2. Implement Moving Average forecasting technique
3. Implement Holt-Winter forecasting technique
4. Implementation of regression for forecasting
5. Measure the Theil's coefficient for suitable data

UNIT IV **11+ 6 Hours**

Six Sigma: Introduction, origin, 3-Sigma Vs Six-Sigma process, cost of poor quality, sigma score, industry applications, six sigma measures, DPMO, yield, sigma score, DMAIC methodology, Six Sigma toolbox
Statistical Process Control, Design of Experiments, Theory of Constraints (TOC

Practical Components

1. Illustrate difference between 3-sigma vs six-sigma process
2. Implementation of DMAIC method with suitable scenario
3. Case study on Enhancing Employee Management using Lean Six Sigma toolbox

48+32 Hours

Textbooks:

1. Jaiwei Ham and Micheline Kamber, Data Mining concepts and techniques, Kauffmann Publishers 2006
2. Efraim Turban, Ramesh Sharda, Jay E. Aronson and David King, Business Intelligence, Prentice Hall, 2008.
3. W.H.Inmon, Building the Data Warehouse, fourth edition Wiley India pvt. Ltd. 2005
4. Marc J. Schniederjans, Dara G. Schniederjans, Christopher M. Starkey, "Business Analytics Principles, Concepts, and Applications with SAS", Associate Publishers,2015.

References:

1. S. Christian Albright, Wayne L. Winston, "Business Analytics - Data Analysis andDecision Making", 5th Edition, Cengage, 2015.
2. U Dinesh Kumar, "Data Analytics", Wiley Publications, 1st Edition, 2017.
3. Ralph Kimball and Richard Merz, The data warehouse toolkit, John Wiley, 3rdedition,2013.
4. Michel Berry and Gordon Linoff, Mastering Data mining, John Wiley and Sons Inc,2nd Edition, 2011
5. https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/
6. https://leansixsigmatoolbox.com/2023/06/08/ongoing-education-lean-six-sigma-employee-engagement/

Internal Assessment Pattern

Cognitive Level	**Int. Test 1 (%)**	**Int. Test 2 (%)**	**Open Book Exam**
Remember	30	30	-
Understand	50	50	-
Apply	20	20	50
Analyze	--	--	50
Evaluate	--	--	-
Create	--	--	-
Total (%)	**100**	**100**	**100**

Remember

1. What is business intelligence?
2. Define predictive analysis

Understand

1. Explain the k-fold cross validation with a suitable machine learning algorithm
2. Explain forecasting techniques for time-series data

Apply

1. Illustrate difference between 3-sigma vs six-sigma process
2. Implementation of DMAIC method with suitable scenario

21CSC22 Web Application Developments Framework

3 0 2 4

Course Outcomes

1. Understand the fundamentals of web framework.
2. Classify model, view and controller layers of a web application.
3. Design a web application using a framework.
4. Know the concept of Java web framework.
5. Understand and analyze how modern-day web applications are different from web sites.
6. Learn the technologies of Python web framework.

COs–POs Mapping

COs	PO$_1$	PO$_3$	PO$_4$
1	3	3	2
2	3	2	2
3	3	3	3
4	3	3	2
5	3	2	3
6	2	3	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

UNIT- I **10+4 Hours**

Fundamentals of Web Framework

Web framework-History, Types of framework architectures, Model–view–controller (MVC), Three-tier organization

Introduction to frameworks-Framework applications, General-purpose website frameworks-Server-side, Client-side features

MVC, Three-tier organisation, Framework

Practical Components

1. Realization of separation of data, data fetch and representation logics
2. Realization of markup- language and it's styling

UNIT- II **14+10 Hours**

Angular – JavaScript web framework

Introduction – Angular MVC, Model, View, Controller, Ajax, Data binding

Angular concepts - Directives, Scopes, Controllers, Modules, Expressions

Developing a simple To-Do application using AngularJS (Developing a single page application)

MVC, Ajax, Data binding

Practical Components

1. Implementation of angular directives
2. Implementation of angular components
3. Implementation of custom/user-defined directive
4. Create a TODO Application using Angular JS
5. Implement routing and navigation in Angular JS

Unit III **13+12 Hours**

React framework

Introduction to React: What is Full-Stack Web Development?, Node.js and NPM, Front-end JavaScript Frameworks and Libraries Overview, Introduction to React, React App Overview, Introduction to JSX, React Components, React Components: State and Props, React Components: Lifecycle Methods Part 1

React Router and Single Page Applications: Presentational and Container Components, React Components: Lifecycle Methods Part 2, Functional Components, React Virtual DOM, React Router, Single Page Applications, React Router: Parameters

React Forms, Flow Architecture and Introduction to Redux: Controlled Forms, Uncontrolled Components, The Model-View-Controller Framework, The Flux Architecture, Introduction to Redux, React Redux Forms

Practical Components

1. React Components
2. React Router and Single page applications
3. Controlled Form Validation, Uncontrolled Forms
4. Create a TODO Application using React JS
5. Creating a movie list Application in React JS
6. Creating a Progressive web application with Reusable React Components

Unit IV **11+6 Hours**

Django – Python web framework

Introduction to Django- History-Django Components-Alternate Components-MVC Architecture in Django
MVC creation in Django – Configuring Django, Creating model, view and controller in Django, REST in Django and templates

MVC, Django, REST

Practical Components

1. Creating models for database queries
2. Writing Django template for rendering data
3. Implementation of REST API using Django

Total: 48+32 Hours

Textbook (s)

1. Angular: Up and Running, Shyam Seshadri, 1st Edition, O'Reilly, 2018
2. Struts the Complete Reference, James Holmes, 2nd Edition, Mc. Graw Hill Professional, 2006.
3. Programming with Django, Wiley Publishing

Reference (s)

1. Angular 6 for Enterprise-Ready Web Applications, Doguhan Uluca, 1st edition, 2018
2. The Definitive Guide to Django, Adrian Holovaty, Jacob Kaplan-Moss, Apress, 2009.
3. Struts 2 In Action, Donald Brown, Chad Michael Davis, Scott Stanlick, Dreamtech press, 2008.

Internal Assessment Pattern

Cognitive Level	**Int. Test 1 (%)**	**Int. Test 2 (%)**	**Assignment Test (%)**
Remember	20	10	--
Understand	50	40	--
Apply	30	30	--
Analyze	--	20	--
Evaluate	--	--	--
Create	--	--	--
Total (%)	100	100	--

Sample Question (S)

Remember

1. Explain model, view and controller
2. List any 4 directives in Angular
3. List the Struts Tag Libraries.
4. Define a scope in Angular
5. Define a template in Django

Understand

1. Write a template in Django that conditionally renders data.
2. Write an example custom tag in Struts
3. Illustrate form validation in Struts.
4. Illustrate the use of ng-filter directive in AngularJS
5. Write an example snippet for ng-if directive in AngularJS
6. Differentiate between plain-validator and field-validator in Struts.
7. For a single Struts application, can we have multiple struts-config.xml files?

Apply

1. Write a snippet that uses ng-for, ng-if and compare it with usage of ng-filter directives in AngularJS
2. Write a sample code for creating a User model in Django that has the following properties:
   a. Name (string)
   b. Age (number
   c. Email (email)
   d. Phone number (number)
3. Write a sample Django template that renders the User data whose age is > 15.
4. Create an action Form bean in Struts.
5. Ilustrate the steps required for setting up validator framework in Struts

Analyze

1. What does the following code snippet in struts-config.xml do?

```
<exception
    key="stockdataBase.error.invalidCurrencyType"
    path="/AvailbleCurrency.jsp"
    type="Stock.account.illegalCurrencyTypeException"
    >
</exception>
```

2. When can global scope be used over local scope in AngularJS?
3. Why is ng-filter more preferrable than ng-if in AngularJS?
4. What happens to existing data in the database when models are edited in Django?
5. Write User and Group models with relevant attributes, where a User can be part of multiple groups. Ensure that when a User is queried, the groups he belongs too are also part of the response.

21MLC32 Cloud Services using AWS

3 0 2 4

Course Outcomes

1. Comprehend different services provided AWS
2. Understand Identify and Access Management in AWS
3. Demonstrate AWS Directory services and AWS Artifact
4. Make use of Amazon Elastic Compute Cloud (EC2) to scalable computing capacity
5. Understand about Elastic File Systems and its features
6. Describe about AWS Storage Options

COs–POs Mapping

COs	PO_2	PO_3	PO_4	PO_5	PO_{12}
1	2	3	2	1	2
2	3	2	1	2	3
3	3	3	2	3	1
4	2	1	3	1	2
5	3	3	1	3	2
6	1	2	3	2	1

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit-I **12+8 Hours**

Introduction to AWS:

Classic Data Center , Virtualization, Service Comparison: AWS, Azure, and GCP, Amazon Web Services (AWS) and its Benefits, AWS Global Infrastructure, Signup an AWS Free Tier Account.

Different Amazon Web Services, Ways to access AWS: CLI, Console, and SDKs, Explore Management Console and Configure AWS CLI

AWS CloudShell

Practical Components

1. Signup an AWS Free Tier Account
2. Explore Management Console and configure AWS CLI
3. Access AWS: CLI, Console, and SDKs
4. Working with AWS services in AWS CloudShell

Unit-II **12+8 Hours**

Identity and Access Management in AWS:

Identity and Access Management (IAM), Managing Users with IAM, Managing Permissions with Groups, IAM Policy and its Elements, IAM Roles ,Password Policy, Best Practices for IAM

AWS Security Token Services, AWS Single Sign-on, AWS Resources Access Manager, Active directory, Microsoft Active directory, AWS Directory Services, AWS Artifact, AWS Audit Manager, Accessing Billing and Alerts.

AWS Certificate Manager

Practical Components

1. Creating New Users, Roles, and Policies on AWS Console
2. Creating an user group for a collection of IAM users
3. Login to AWS Console via MFA
4. AWS Artifact

Unit-III **12+8 Hours**

Amazon Elastic Compute Cloud (EC2):

Amazon Elastic Compute Cloud (EC2) and Its Benefits, Amazon Machine Image (AMI), Security Groups in AWS, Authentication through Key-pair, Hardware Tenancy – Shared vs. Dedicated, Networking Layer in EC2: VPC, Elastic Network Interface (ENI) and Its Attributes

Public IP vs. Elastic IP, Instance Store

Elastic Block Store (EBS), Its Features and Volume Types

Solid State Drive: General Purpose SSD and Provisioned IOPS

Hard Disk Drive: Throughput Optimized HDD and Cold HDD

Snapshots, Elastic File System (EFS) and Its Features, Amazon Batch, Windows File Server

Amazon FSx for Windows File Server

Practical Components

1. Creating an EC2 instance and Custom AMI
2. Host the Website Inside EC2 instance
3. Create an Elastic IP and attaching an EBS Volume Externally
4. Creating a Snapshot

Unit-IV **12+8 Hours**

AWS Storage Options:

Traditional Storage, Need to Move to Cloud Storage, Traditional vs. Cloud Storage Cost, Cloud Storage, Different Storage Options Available on AWS, Simple Storage Service (S3) and Its Components, Working of S3, Bucket Policy, Access Control List (ACL)

Cross-Region Replication (CRR) and Its Use Case, Amazon S3 Transfer Acceleration, Choice of Storage Classes on S3, Lifecycle Policy of S3 Bucket, AWS Backup.

CDN: CloudFront, Storage Gateway

Practical Components

1. Hosting a Static Website on Amazon S3
2. Replicating Data Across Regions
3. Transfer and Retrieve Data from Glacier
4. Accessing Website Using CloudFront

48+32 Hours

Textbook(s):

1. Amazon Web Services (AWS) 2022 Everything You Need To Know About The Amazon Web Service From Start To End, 2022, Maxwell Harris, Maxwell Harris.
2. Implementing AWS: Design, Build, and Manage your Infrastructure, Yohan Wadia, Rowan Udell, Lucas Chan, Udita Gupta, packt publishers.

Internal Assessment Pattern

Cognitive Level	Internal Test 1 %	Internal Test 2 %	Lab Examination %
Remember	25	25	---
Understand	50	25	---
Apply	25	50	40
Analyze	---	---	20
Evaluate	---	---	40
Create			
Total (%)	**100**	**100**	**100**

Sample Question(s)

Remember

1. List any two cloud services provided by Amazon AWS.
2. What is AWS Artifact?
3. Define EC2.
4. What is Access Control List?

Understand

1. What is Virtualization and explain about that.
2. Explain about Identity and Access Management (IAM).
3. Write and explain about AWS Directory Services.
4. Explain about Access Control List (ACL).

Apply

1. How do we sign up for an AWS Free Tier Account?
2. Illustrate the process of Move to Cloud Storage from normal storage.
3. How Cross-Region Replication (CRR) works?

21CS007 Cloud Computing Essentials (Elective III)

3 0 2 4

Course Outcomes

1. Interpret the architecture and infrastructure models of cloud computing, strengths, and limitations of cloud computing.
2. Understand the virtualization concepts of machines and data centers.
3. Infer the design concepts of cloud ready applications
4. Compare different cloud centre's implementation
5. Understand the concepts of cloud scaling and disaster recovery
6. Analyze the performance, scalability, and availability of the underlying cloud technologies and software

CO–PO Mapping

CO	PO_2	PO_5	PO_6	PO_7	PO_8
1	3	3	2	3	1
2	3	1	1	2	3
3	2	2	3	1	2
4	2	1	3	3	1
5	2	1	3	1	2
6	2	3	3	3	3

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12+6 Hours**

Characterization of Distributed Systems: Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges. System Models: Introduction, Architectural Models- Software Layers, System Architecture, Variations, Interface and Objects, Design Requirements for Distributed Architectures, Fundamental Models- Interaction Model, Failure Model, Security Model.

Communication between Distributed Objects- Object Model, Distributed Object Modal.

Practical Components:

1. Study of Cloud Computing & Architecture.
2. Install KVM/VMware Workstation for creating a virtual environment on windows platform. Concept: Virtualization
3. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.

Unit II **12+8 Hours**

Introduction to Cloud Computing

Overview of Computing Paradigm: Recent Trends in Computing, Evolution of Cloud Computing. Introduction to Cloud Computing: Cloud Computing (NIST Model), Properties, Characteristics & Disadvantages, Role of Open Standards. Cloud Computing Architecture: Cloud Computing Stack, Service Models (XaaS), Deployment Models. Infrastructure as a Service (IaaS): Introduction to IaaS, Resource Virtualization. Platform as a Service (PaaS): Introduction to PaaS, Cloud Platform and Management. Software as a Service (SaaS): Introduction to SaaS, Web services, Web 2.0, Web OS.

Companies in the Cloud Today, Amazon Web Servies, Google services, IBM Cloud, Windows Azure, Tata Cloud, Salesforce.com

Practical Components:

1. Install Google App Engine. Create hello world app and other simple web applications using python/java.
2. Use GAE launcher to launch the web applications.
3. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
4. Signup an AWS Free Tier Account and Explore Management Console and configure AWS CLI

Unit III **12+10 Hours**

Virtualization & Design

Virtualization, Virtual machine, Implementation Levels of Virtualization, Virtualization Structures/Tools and Mechanisms, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management, Data centre, Virtualization for Data-Centre Automation. Service Levels for Cloud Applications Ready for the cloud: Web Application Design, Machine Image Design, Privacy Design, Database Management.

various hypervisors like VMware, KVM, oracle VM, HTTPS,Electronic Payment

Practical Components:

1. Working on to transfer the files from one virtual machine to another virtual machine.
2. Working on to launch virtual machine using trystack (Online Openstack Demo Version)
3. Set up a pseudo-distributed, single-node Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux. After successful installation on one node, configuration of a multi-node Hadoop cluster(one master and multiple slaves).Hadoop file management: Adding files and directories, Retrieving files, Deleting files
4. Run Wordcount application using Hadoop single node cluster.
5. Move files between regular Linux file-system and HDFS using put and get commands

Unit IV

Cloud Service Providers **12 + 8 Hours**

EMC, EMC IT, Captiva Cloud Toolkit, Google, Cloud Platform, Cloud Storage, Google Cloud Connect, Google Cloud Print, Google App Engine, Amazon Web Services, Amazon Elastic Compute Cloud, Amazon Simple Storage Service, Amazon Simple Queue, service, Microsoft, Windows Azure, Microsoft Assessment and Planning Toolkit, SharePoint, IBM, Cloud Models, IBM Smart Cloud, SAP Labs, SAP HANA Cloud Platform, Virtualization Services Provided by SAP, Sales force, Sales Cloud.

Service Cloud: Knowledge as a Service, Rack space, VMware, Manjra soft, Aneka Platform

Practical Components:

1. Working and installation of Microsoft Azure
2. Working with Mangrasoft Aneka Software
3. Installation and Configuration of Justcloud
4. Working with AWS services in AWS CloudShell

Total: 48+32 Hours

Textbook (s)

1. George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems- Concepts and Design”, Fourth Edition, Pearson Publication
2. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
3. Cloud Computing: Principles and Paradigms, Editors: RajkumarBuyya, James Broberg, Andrzej M. Goscinski, Wile, 2011
4. Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
5. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010

Reference (s)

a) Michael Miller, Cloud Computing-Web Based Applications that change the way you work and collaborate online, 1st Edition, Pearson Education, Publishing, 2011

b) Kai Hwang, Geoffrey C Fox and Jack J.Dongarra, Distributed & Cloud Computing from Parallel Processing to the Internet of Things , 1st Edition, MK Publishing, 2010

c) David S Linthicum, Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide, 1st Edition, Addison-Wesley, 2009

d) Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, SubraKumaraswamy, ShahedLatif, O’Reilly, SPD, rp2011.

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)
Remember	30	30

Understand	40	40
Apply	30	30
Analyze	--	--
Evaluate	--	--
Create	--	--
Total (%)	100	100

SAMPLE QUESTION (S)

Remember

1. Define Cloud Computing.
2. List types of virtualizations.
3. Define proactive scaling.
4. What is CIATriad?

Understand

1. Explain different cloud Infrastructure models.
2. Explain different levels of virtualization.
3. Explain about cloud Network security in detail.
4. Explain about Recovery Point Objective.
5. With neat diagram explain the functioning of Xen Architecture.
6. List and explain various cloud service providers risks.

Analyze

1. Compare cloud center and service infrastructure.
2. Analyze different cloud services provided by Amazon

21CS008 Cryptography and Network Security (Elective III)

3 0 2 4

Course Outcomes

1. Explain the fundamentals of Cryptography, encryption and decryption algorithms
2. Make use of the symmetric and public key cryptographic algorithms
3. Choose the various authentication applications for security
4. Interpret the functionalities of IP and web security.
5. Demonstrates the functionalities of firewalls
6. Explain various non-cryptographic protocol vulnerabilities

CO–PO Mapping

COs	PO3	PO6	PO8
1	2	2	3
2	2	2	3
3	2	1	3
4	2	1	2
5	2	2	3
6	2	2	2

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Unit I **12+10 Hours**

Introduction: Security Attacks, Security Goals, Computer criminals, Methods of Defence, Security Services, Security Mechanisms. **Basics of Cryptography:** Symmetric Cipher Model, Substitution Techniques, Transportation, Techniques, Other Cipher Properties- Confusion, Diffusion, Block and Stream Ciphers. Block Cipher Design Principles and Modes of Operations, **Symmetric Key Cryptosystems:** Principles of Private Key System, Data Encryption Standard (DES), Strength of DES, Triple DES, International Data Encryption Algorithm(IDEA), Advanced Encryption Standard (AES)

Blowfish, CAST-128.

Practical Components:

1. Write a program that contains a string (char pointer) with a value "Hello world". The program should XOR each character in this string with 0 and displays the result.
2. Write a program that contains a string (char pointer) with a value "Hello world". The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a program to perform encryption and decryption using the following algorithms
   a. Ceaser cipher
   b. Substitution cipher
4. Write a program to perform encryption and decryption using the following algorithms
   a. Mono-alphabetic cipher
   b. Hill Cipher
5. Write a program to perform encryption using the following algorithms
   a. Railfence Technique
   b. Transposition Technique

Unit II **12+8 Hours**

Public Key Cryptography: Principles of Public Key Cryptosystems, RSA Algorithm, Diffie-Hellman Key Exchange. Cryptographic Hash Functions: Principles of Cryptographic Hash functions, Applications of Cryptographic Hash Functions, Secure Hash Algorithm (SHA), Message Authentication Codes – Message Authentication Requirements and Functions, HMAC, Digital Signatures, Elgamal Digital Signature Schemes

Digital Signature Standards

Practical Components:

1. Write a program to implement the Play-fair Cipher algorithm.
2. Write a program to implement the DES algorithm logic.
3. Write a program to implement the Blowfish algorithm logic.

Unit III **12+8 Hours**

Authentication Applications: Kerberos, Key Management and Distribution, X.509 Directory Authentication service, Public Key Infrastructure, Electronic Mail Security: Pretty Good Privacy, S/MIME. **IP Security:**

Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining security Associations, Internet Key Exchange,

Web Security: Web Security Considerations, Secure Sockets Layer and Transport Layer Security,.

HTTPS, Electronic Payment

Practical Components:

1. Write a Java program to implement RSA algorithm.
2. Calculate the message digest of a text using the SHA-1 algorithm.
3. Write a program to implement the Triple-DES algorithm logic.
4. Write a program to implement the AES algorithm logic.

Unit IV

Locking, Recovery Systems, Indexing, Different Types of Data **12 + 6 Hours**

IDS and Firewalls: Intruders, Intrusion Detection, Password Management, Firewalls-Characteristics, Types of Firewalls, Placement of Firewalls, Firewall Configuration, Trusted Systems. Non-cryptographic protocol Vulnerabilities: DoS, DDoS, Session Hijacking and Spoofing, Software Vulnerabilities- Phishing, Buffer Overflow.

Format String Attacks, SQL InjectionCybercrime and Computer Crime, Intellectual Property.

Practical Components:

1. Implement the Diffie-Hellman Key Exchange mechanism.
2. Write a program that can encrypt and decrypt using a general substitution block cipher.
3. Write a program that can encrypt and decrypt in 4-bit in cipher clock chaining mode using DES.

Total: 48+32Hours

Textbook (s)

1. William Stallings, "Cryptography And Network Security – Principles and Practices", 7th edition, Pearson Education Limited 2017.
2. AtulKahate, "Cryptography and Network Security", 2nd edition Tata McGraw-Hill, 2003.
3. Behourz A Forouzan, Cryptography and Network Security, 2nd edition, Tata McGraw-Hill 2011

Reference (s)

1. Matt Bishop ,"Computer Security art and science ", Second Edition, Pearson Education, 2002
2. Wade Trappe and Lawrence C. Washington, "Introduction to Cryptography with Coding Theory" Second Edition, Pearson Education, 2007
3. Jonathan Katz, and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press, 2007
4. Douglas R. Stinson, "Cryptography Theory and Practice", Third Edition, Chapman & Hall/CRC, 2006
5. Wenbo Mao, "Modern Cryptography – Theory and Practice", Pearson Education, First Edition, 2006.
6. OWASP top ten security vulnerabilities: http://xml.coverpages.org/OWASPTopTen.pdf

Internal Assessment Pattern

Cognitive Level	Int. Test 1 (%)	Int. Test 2 (%)
Remember	30	30
Understand	40	40
Apply	30	30
Analyze	--	--
Evaluate	--	--
Create	--	--
Total (%)	100	100

SAMPLE QUESTION (S)

Remember

1. Mention any two security attacks
2. List any two goals of security
3. Define Hash function

Understand

1. Differentiate between asymmetric and symmetric key cryptography
2. How do we achieve authentication?
3. Differentiate between the two applications of hash function.

Apply

1. How do we apply PGP to the Email Security?
2. Implement firewall using iptables command.
3. Can message encryption itself provide measure of authentication?

Open Book Exam Questions

Question 1:

A man named Jones wanted to chat with his girlfriend Goldie. But he can see that all his family is around him and even his girlfriend is also with her parents. So, he thought to send a secret message to his girlfriend. They usually love to meet in the "Central Park". Now Jones wants to send a message to Goldie as "Hi Goldie. How are you. Because you are with your parents, and I am with my parents we cannot speak with each other. But I want to meet you at our favourite place central park tomorrow after your class". Help jones to convert the message to unreadable format using the key of their favorite place. Also suggest him how would jones tell Goldie that the letter is originated from Jones only.

Question 2:

You have decided to start a startup after graduation. But you alone cannot be doing this, so you have to ask help from your friends. Suggest what techniques you require and which sort of people you would select so that your company will be a huge success. Explain the techniques clearly to your friends and tell them what they have to do in detail.

21DS606 Optimization techniques for Machine Learning Lab

0 0 3 1.5

Course Outcome

1. Implementation of basic classification & regression ML models
2. Practical on Constrained Optimization on linear and non-linear programming
3. Practical on Convex Optimization with suitable python implementation
4. Practical implementation of large Scale Optimization
5. Practical implementation of Regularization and Sparsityoptimization
6. Implementation of deep learning architectures in python

CO–PO Mapping

CO	PO4	PO5
1	3	2
2	3	3
3	3	3
4	2	2
5	2	2
6	1	1

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Lab Manual

1. **Optimization for Machine Learning Lab**– AIDS Department, GMRIT.

Textbook (s)

1. "Optimization for Machine Learning" by SuvritSra, Sebastian Nowozin, and Stephen Wright
2. "Convex Optimization" by Stephen Boyd and LievenVandenberghe
3. "Numerical Optimization" by Jorge Nocedal and Stephen J. Wright

Reference (s)

1. "Stochastic Gradient Descent and Beyond" by Martin Jaggi
2. "Sparse Modeling: Theory, Algorithms, and Applications" by Irina Rish and GenadyGrabarnik

Lab Experiments

1. Implementation of Regression problem for a sample dataset
2. Implementation of Binary and Multiple-Class Classification for a sample datasets
3. Understanding and implementation of Gradient descent.
4. Solving Newton’s Method for optimization using Python programming.
5. Implementing and Interpreting Stochastic Gradient Descent
6. Solving a Linear and Non-Linear Programming problem with suitable dataset
7. Convex Optimization in Python using CVXPY and PULP
8. Illustration and implementation of Mini Batch gradient descent.
9. Implementation Elastic net Regression with L1 and L2 penalties
10. Implementing conjugate gradient method with python
11. Implementation and solution of Quasi Newton’s method for advanced optimization.
12. Illustration of Artificial Neural network and implementing MNIST dataset
13. ANN with back propagation using ReLu and Adam.
14. Implementing Convolution neural networks (cat vs dog dataset)
15. Implementing LSTM neural **network for textual** sentimental analysis.
16. Implement the Variational Auto-encoder in neural network.

Augmented experiments

1. Comparison of Optimization Algorithms
2. Hyper parameter Optimization
3. Regularization Techniques Comparison
4. Optimizing Neural Network Architecture
5. Large-scale Optimization for Deep Learning
6. Online Learning with Optimization
7. Optimization for Sparse Data
8. Optimization for Deep Reinforcement Learning

21MPX01 Mini Project

0 0 3 1.5

Course Outcomes

1. Identify a contemporary engineering application to serve the society at large
2. Use engineering concepts and computational tools to get the desired solution
3. Justify the assembled/fabricated/developed products intended
4. Organize documents and present the project report articulating the applications of the concepts and ideas coherently
5. Demonstrate ethical and professional attributes during the project implementation
6. Execute the project in a collaborative environment

COs – POs Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1	3	2	-	-	-	3	2	-	-	-	-	-
2	3	3	-	-	3	-	-	-	-	-	-	-
3	3	3	3	2	-	-	-	-	-	-	2	-
4	-	-	-	-	-	-	-	-	-	3		2
5	-	-	-	-	-	-	-	3	-	-	-	-
6	-	-	-	-	-	-	-		3	-	-	-

3–Strongly linked | 2–Moderately linked | 1–Weakly linked

Mini Project: The curriculum offers Mini Projects in two different forms viz: (i) Mini Project as a mandatory component in all lab courses (ii) 2 credit Mini Project during 5th or 6th semester. With respect to second one (ii) student will take mini project batch wise and the batches will be divided as similar to lab courses. The report will be evaluated by a committee as nominated by CoE constituted with internal & external panels

21ESX02 Employability Skills II

0 0 2 2

Course Outcomes

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Introspect & develop life skills with constructive approach
3. Assess and improve analytical skills
4. Choose appropriate Supervised Learning techniques for a given dataset
5. Apply different clustering techniques to characterize subgroups.
6. Communicate results in terms relevant to science, business etc.

COs – POs Mapping

COs	PO1	PO2	PO3	PO5	PO6	PO 8	PO10	PO 12
CO1	-	-	-	-	-	-	3	-
CO2	-	-	-	-	3	-	2	2
CO3	2	-	-	-	-	2	-	-
CO4	3	-	3	2	-	2	-	-
CO5	3	2	3	2	-	-	-	2
CO6	2	2	3	2	3	-	-	-

3–Strongly linked | 2–Moderately linked| 1–Weakly linked

Soft Skills:

Sl No.	Topic & Content	No. of Periods
1.	**Resume (Recap):** Resume? Templates? Mistakes to be avoided in a Resume and Steps to be followed in preparing it.	**01**
2.	**Group Discussions (Recap) & Practice:** GD? Stages of a GD, Skills assessed in a GD, Blunders to be avoided, How to excel in a GD? Practice sessions and sharing Feedback. (Screening sample Videos)	**01**
3.	**Interview Skills:** Interview? Types of Interviews, Dos & Don'ts, Skills assessed in an Interview, Mistakes to be avoided, How to equip oneself to excel? How to handle the Typical Interview Questions? (with Examples)	**03**
4.	**Mock Interviews**: Practice sessions with Feedback.	**02**
5.	**Exercises related to Communication:** Email Writing, Voice Versant., etc.	**01**
	Total Periods	**08**

Quantitative Aptitude:

Sl. No.	Topic	No. of Periods
1.	Time and Distance	01
2.	Time and Distance	01
3.	Problems on Trains	01
4.	Problems on Trains	01
5.	Blood relations	01
6.	Ratio and Proportions	01
7.	Calendars	01
8.	Clocks	01
	Total Periods	**08**

Domain Specific:(Data Mining using WEKA)

Sl. No	Topic	No. of Periods
1.	**Classification of Data** General Approach to Solving Classification Problem-Decision Tree Induction-Regression-Nearest-NeighborClassifiers-Bayesian Classifiers	3
2.	Implement the Classification using Decision Tree Induction Algorithm. (Practical)	2
3.	Implement the Classification using Regression (Practical)	2
4.	Implement the Bayesian Classification Algorithm(Practical)	1
5.	**Cluster analysis & Mining Complex Types of Data** Cluster analysis -K-means-Agglomerative Hierarchical Clustering	4
6.	1. Implement K-means clustering Algorithm. (Practical) 2. Implement Hierarchical clustering Algorithm (Practical)	3
	Total Periods	**15**

Textbook (s)

1. Pang, Ning Tan, Michael Steinbach, Vipin Kumar Introduction to Data Mining, 3rd Edition, Pearson Addison Wesley, 2016
2. Jiawei Han &MichelineKamber, Data Mining Concepts and Techniques, 3rd Edition, Morgan Kaufmann, 2012

Reference (s)

1.Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann Publishers, 2011
2. Margaret H Dunham, Data Mining Introductory and advanced topics, 2nd Edition, Pearson Education, 2004
3. Arun K Pujari, Data Mining Techniques, 2nd Edition, University Press, 2099.

21HSX12 CC & EC Activities II

0 0 1 1

Course Outcomes

1. Interpret and present the abstractive technical information through an activity
2. Think critically in providing solutions to the generic and common problems
3. Demonstrate the creative thinking in dealing with liberal arts
4. Instill team sprit through active engagement with the peer
5. Develop programs of common interest having social impact
6. Empower the under privileged through motivational activities

COs – POs Mapping

COs	PO6	PO7	PO9	PO10
1	-	-	-	3
2	3	2	-	-
3	3	-	-	-
4	-	-	3	-
5	3	-	-	-
6	3		-	-

Co-Curricular and Extra Curricular (CCEC) Activities:

Students shall acquire 1 credit each in 2nd and 3rd years with the following scheme:

Scheme of evaluation for the CCEC activities:

- No. of slots in each Semester @ 2 slots every week : 24
- No. of Stream (2-CC + 1-EC) : 3
- No. of slots allotted for each stream : 8

Requirement for the award of 1- Credit

- Students shall choose at least two streams of events in each semester
- Students shall secure 75% attendance in each stream of events to obtain a certificate
- Students shall obtain 2 certificates of Participation in each semester.

The credits earned through these courses will be indicated in the grade sheet and will not beaken into account for CGPA calculation.

AUDIT COURSE

Course Outcomes

At the end of the Audit Course the students will be able to

CO1: Interpret the meaning of values and select their goals by self- Investigation based on personal values.

CO2: Interpret the major events and issues related to a period in Indian history.

CO3: Assess the benefits and limitations of science and its application in technological developments towards human welfare

CO4: Check the awareness regarding basic human rights and to uphold the dignity of every individual.

CO5: Assess the individual and group behaviour and understand the implications of organizational behaviour on the process of management.

CO6: Determine the appropriateness of various leadership styles and conflict management strategies used in organizations.

Cos-POs Mapping:

COs	PO12
CO1	3
CO2	3
CO3	3
CO4	2
CO5	3
CO6	2