GMR Institute of Technology
An Autonomous Institute Affiliated to JNTU-GV

GMRIT
Training Tomorrow's Engineers Today

1.1.3 (A) Syllabus copy of the courses highlighting Focus on Employability/Entrepreneurship/ Skill development along with their course outcomes

Department of Computer Science and Engineering

INDEX

The Vision of GMRIT

- To be among the most preferred institutions for engineering and technological education in the country.
- 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 very best.
- An institution where ethics, excellence and excitement will be the work religion, while research, innovation and impact, the work culture.

The Mission of GMRIT

- To turnout disciplined and competent engineers with sound work and life ethics
- To implement outcome based education in an IT-enabled environment
- To encourage all-round rigor and instill a spirit of enquiry and critical thinking among students, faculty and staff
- To develop teaching, research and consulting environment in collaboration with industry and other institutions

Department Vision

To be a preferred department of learning for students and teachers alike, with a commitment towards Academics & Research, serving the students in an atmosphere of innovation, critical thinking and making them Industry ready.

Department Mission

1. To provide adaptable education in a collaborative and innovative environment in skilling the graduates to solve real world problems in the field of Computer Science and Engineering.
2. To prepare the students as critical thinking professionals with multidisciplinary research orientation and Innovation
3. To instil ethical values and nurture the graduates who will be able to contribute to society.

Program Educational Objectives

1. Employ logical and analytical skills in solving complex real-world engineering problems in the areas of computer Science and allied fields.
2. Adaptable to emerging technologies with enhanced professional skills and ability towards continuous learning, facilitating higher studies and research
3. Demonstrate professional ethics, leadership qualities and promote inclusive and collaborative growth with human values towards societal interest

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 of solutions)

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)

PO10: 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)

PO11: 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)

PO12: 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 the process and service industries apart from the community utilities. (Program Specific)

PSO2: Ability to design, develop and implement management systems, E-Commerce tools and Web Apps for product development. (Program Specific)

Department of Computer Science and Engineering

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

First Semester							
No	**Course Code**	**Course Name**	**POs**	**L**	**T**	**P**	**C**
1	21HSX01	Communicative English	1,6,,10,11,12	2	-	-	2
2	21MAX01	Engineering Mathematics I	1,2,3,12,PSO1,PSO2	3	-	-	3
3	21PYX01 21CYX01	Engineering Physics / Engineering Chemistry	1,2,3,6,7,12 / 1,2,3,6,7,12	3/3	-	-	3/3
4	21BEX01 21BEX06	Basics of Engineering / IT Workshop	1 ,2,3,6,7,12 PSO1,PSO2/ 3,4,5,6,7,8,9,10,11,12	3/ -	-	-/ 3	3/1.5
5	21BEX02	Problem Solving and Programming Skills	1,2,3,12,PSO1	3	-	-	3
6	21BEX03	Problem Solving and Programming Skills Lab	1,2,3,4,12,PSO1	-	-	3	1.5
7	21BEX04/ 21BEX05	Engineering Drawing / Engineering Workshop	1,2,3,4,5,9,10,11,12/ 1,2,3,4,9,10,11,12	-	-	3/3	1.5/1.5
8	21PYX02/ 21CYX02	Engineering Physics Lab /Engineering Chemistry Lab	1,2,3,4,5,9,11,12/ 1,2,3,4,5,7,9,11,12	-	-	3/3	1.5
9	21HSX02/-	Communicative English Lab/-	1,4,5,9,10,11,12	-	-	3/-	1.5/-
			Total	**14/11**	**-**	**12/12**	**20/17**
Second Semester							
1		Language Elective	6,9,10,11,12	2	-	-	2
2	21MAX02	Engineering Mathematics II	1,2,3,12, PSO1,PSO2	3	-	-	3
3	21CYX01/ 21PYX01	Engineering Chemistry /Engineering Physics	1 ,2,3,6,7,12/ 1 ,2,3,6,7,12	3/3	-	-	3/3
4	21BEX01/ 21BEX06	Basics of Engineering/ IT Workshop	1 ,2,3,6,7,12 PSO1,PSO2/ 3,4,5,6,7,8,9,10,11,12	-/3	-	3/-	1.5/3
5	21BEX07	Python Programming	1,2,3,12,PSO1	3	-	-	3
6	21BEX08	Python Programming Lab	1,2,3,4,5,12,PSO1	-	-	3	1.5
7	21BEX05/ 21BEX04	Engineering Workshop / Engineering Drawing	1,2,3,4,5,9,10,11,12/ 1,2,3,4,9,10,11,12	-	-	3/3	1.5/1.5
8	21CYX02/ 21PYX02	Engineering Chemistry Lab/Engineering Physics Lab	1,2,3,4,5,9,11,12/ 1,2,3,4,5,7,9,11,12	-	-	3/3	1.5/1.5
9	-/21HSX02	-/Communicative English Lab	1,4,5,9,10,11,12	-	-	-/3	-/1.5
			Total	**11/14**	**-**	**12/12**	**17/20**
Third Semester							
1	21MA304	Probability and Statistics using Python	1,2,4,10,12	3	-	2	4
2	21HSX10	Engineering Economics and Project Management	1,2,10,11,PSO2	3	-	-	3
3	21CS303	Data Structures	1,2,3,12,PSO2	3	-	-	3
4	21CS304	Digital Logic Design	1,2,3,4,5,PSO1	3	-	2	4
5	21CS305	Discrete Mathematical Structures	1,2,3,12,PSO1	3	-	-	3
6	21CS306	Object Oriented Programming with JAVA	1,2,3,12,PSO2	3	-	-	3
7	21CS307	Data Structures Lab	1,2,3,4,10,12,PSO1	-	-	3	1.5
8	21CS308	JAVA Lab	1,2,3,4,10,12,PSO2	-	-	3	1.5
9	21ESX01	Employability Skills I	1,2,3,4,5,9,10,11,12	0	-	2	-
10	21HSX11	CC & EC Activities I	9,10,12	-	-	1	-
			Total	**18**	**-**	**13**	**23**
Fourth Semester							
1	21IT304	Database Management Systems	1,2,3,10,12,PSO2	3	-	-	3
2	21IT403	Operating Systems	1,2,12,PSO1,PSO2	3	-	-	3
3	21CS403	Computer Organization and	1,2,3,12,PSO1	3	-	-	3

		Architecture					
4	21CS404	Design and Analysis of Algorithms	1,2,3,4,12,PSO1	3	-	2	4
5	21CS405	Web Coding and Development	1,2, 3, 5, 8,PSO2	3	-	-	3
6	21IT308	Database Management Systems Lab	1,2,3,4,8,10,12,PSO2	-	-	3	1.5
7	21CS407	Web Coding and Development Lab	1,2,3,4,5,8,10,PSO2	-	-	3	1.5
8	21ESX01	Employability Skills I	1,2,3,4,5,9,10,11,12	0	-	2	2
9	21HSX11	CC & EC Activities I	9,10,12	-	-	1	1
			Total	**15**	**-**	**11**	**22**
Fifth Semester							
1	21EC502	Microprocessors and Microcontroller Programming (Integrated)	1,2,3,4,5,PSO1	3	-	2	4
2	21CS502	Artificial Intelligence and Machine Learning	1,2,3,4,12,PSO1,PSO2	3	-	-	3
3	21CS503	Computer Networks (Integrated)	1,2,4,5,7,12,PSO1	3	-	2	4
4	21CS504	Theory of Computation	1, 2,3,12,PSO1	3	-	-	3
5		Elective I (Professional Elective)		3	-	-	3
6		Elective II (Open Elective I)		3	-	-	3
7	21CS507	Artificial Intelligence and Machine Learning Lab	1,2,4,5,12,PSO1,PSO2	-	-	3	1.5
8	21TPX01	Term Paper	1,2,4,5,8,9,10,11,12, PSO1,PSO2	-	-	3	1.5
9	21ESX02	Employability Skills II	1,2,3,4,5,9,10,11,12	0	-	2	-
10	21HSX12	CC & EC Activities II	9,10,12	-	-	1	-
11	21SIX01	Summer Internship I	1,2,3,6,7,8,9,10,11,12 ,PSO1,PSO2	-	-	-	1
			Total	**18**	**-**	**12**	**24**
Sixth Semester							
1	21CS601	Compiler Design	1, 2,3,12, PSO1	3	-	-	3
2	21CS602	Internet of Things	1, 2,3,4 7, PSO1	3	-	-	3
3	21CS603	Software Engineering	1,2,3,5,11,PSO1	3	-	-	3
4		Elective III (Professional Elective)		3	-	2	4
5		Elective IV (Open Elective II)		3	-	-	3
6	21CS606	Case Tools Lab	1,2,3, 4, 5,10,11,PSO1	-	-	3	1.5
7	21MPX01	Mini Project	All POs & PSOs	-	-	3	1.5
8	21ESX02	Employability Skills II	1,2,3 ,4,5,9,10,11,12	0	-	2	2
9	21HSX12	CC & EC Activities II	9,10,12	-	-	1	1
10	21ATX01	Environmental Studies	1,3,6,7	-	-	-	-
11	21ATX02	Professional Ethics and Human Values	----	-	-	-	-
12	21ATX---	Audit Course	----	-	-	-	-
			Total	**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	21SIX02	Summer Internship II	All POs & PSOs	-	-	-	1
5	21PWX01	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**
Language Electives							
1	21HSX03	Advanced Communicative English	6,9,10,11,12	2	-	-	2
2	21HSX04	Communicative German	6,9,10,11,12	2	-	-	2
3	21HSX05	Communicative French	6,9,10,11,12	2	-	-	2
4	21HSX06	Communicative Japanese	6,9,10,11,12	2	-	-	2
5	21HSX07	Communicative Spanish	6,9,10,11,12	2	-	-	2

6	21HSX08	Communicative Korean	6,9,10,11,12	2	-	-	2
7	21HSX09	Communicative Hindi	6,9,10,11,12	2	-	-	2
Elective I :Professional Elective I							
Career Path I, II, III							
1	21CSC11	Exploratory Data Analytics (AI&ML)	1,2,4,10,12,PSO2	3	-	-	3
2	21CSC21	Web Programming Languages (Full Stack Developer)	1,2,3,5,PSO1, PSO2	3	-	-	3
3	21ITC31	Fundamentals of Security (Cyber Security)	1,2,5,PSO1, PSO2	3	-	-	3
Non-Career Path (Core Electives)							
4	21CS004	Principles of Programming Languages	1, 2, 3, 4	3	-	-	3
5	21CS005	Mobile Computing	1,2,3,7	3	-	-	3
6	21CS006	Distributed Operating Systems	1,2,3,12	3	-	-	3
7		MOOCs/Honors		3	-	-	3
Elective III: Professional Elective II							
Career Path I, II, III							
1	21CSC12	Deep Learning (AI&ML)	1,2,3,4,5,12, PSO1	3	-	2	4
2	21CSC22	Web Application Developments Framework (Full Stack Developer)	1,2, 3,5,12,PSO2	3	-	2	4
3	21ITC32	Cybernet Security (Cyber Security)	1,3,4, 5, 12,PSO1,PSO2	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 V : Professional Elective III							
Career Path I, II, III							
1	21CSC13	Natural Language Processing (AI&ML)	1,2,3,4,12, PSO2	3	-	-	3
2	21CSC23	Web Application Databases (Full Stack Developer)	1,2,3,5,12,PSO2	3	-	-	3
3	21ITC33	Cloud Security (Cyber Security)	1,2,4,5,8	3	-	-	3
Non-Career Path (Core Electives)							
4	21IT008	Social Network Analysis	1,2, 3,4, 5	3	-	-	3
5	21CS011	Optimization Techniques	1,2, 3, 5,12	3	-	-	3
6	21CS012	Wireless Adhoc Networks	1,2, 3,10,12	3	-	-	3
Elective VI : Professional Elective IV							
1	21CS014	Green Computing	1,2,3,6,7	3	-	-	3
2	21CS015	Software Project Management	1,2,3,11,12, PSO2	3	-	-	3
3	21CS016	Soft Computing	1,2,3,4,12,PSO1	3	-	-	3
Elective VIII: Professional Elective							
1	21CS017	Fundamentals of Social Network Analysis	1,2,3, 4,5,PSO2	-	-	-	3
2	21CS018	Information Retrieval Systems	1,2,3,4,5,12	-	-	-	3
3	21CS019	Fundamentals of Devops	1,3, 5,11,PSO2	-	-	-	3
4		MOOCs/Honors		-	-	-	3
Open Elective I, II, III, and IV (Elective II, IV, VII, and IX)							
1	21CE001	Disaster Management	2,7	3	-	-	3
2	21EE001	Electrical Installation, Safety and Auditing	2,3,6,8	3	-	-	3
3	21ME001	Fundamentals of Optimization Techniques	1,2,3,5	3	-	-	3
4	21EC001	Sensors for Engineering Applications	1	3	-	-	3
5	21CS001	Fundamentals of Artificial Intelligence	1,2,3	3	-	-	3
6	21CH001	Energy Conversion and Storage Devices	1,3,6,7	3	-	-	3
7	21IT001	Fundamentals of Multimedia	3,5,7	3	-	-	3
8	21BS001	Nano Materials and Technology	1,12	3	-	-	3
9	21DS001	Fundamentals of Data Science	1,2	3	-	-	3
10	21CE002	Air Pollution and Environmental Impact Assessment	6,7,12	3	-	-	3
11	21EE002	Renewable Energy Sources	2,7	3	-	-	3

12	21ME002	Principles of Entrepreneurship	1,5,8,11	3	-	-	3
13	21EC002	Electronics for Agriculture	1,2	3	-	-	3
14	21CS002	Fundamentals of Machine Learning	2,5	3	-	-	3
15	21CH002	Industrial Safety and Hazard Management	1,2,3,6,8	3	-	-	3
16	21IT002	Fundamentals of Cloud Computing	2,6,7,8,12	3	-	-	3
17	21BS002	Advanced Numerical Techniques	1,2	3	-	-	3
18	21BS003	Functional Materials and Applications	1,7	3	-	-	3
19	21CE003	Solid Waste Management	3,7,12	3	-	-	3
20	21EE003	Fundamentals of Electrical Vehicle Technology	2,3,12	3	-	-	3
21	21ME003	Industrial Engineering and Management	1,11	3	-	-	3
22	21EC003	Interfacing and Programming with Arduino	1,2	3	-	-	3
21	21CS003	Data Science for Engineering Applications	2,3,4	3	-	-	3
24	21CH003	Industrial Ecology for Sustainable Development	2,6,7	3	-	-	3
25	21IT003	Fundamentals of Mobile Computing	1,7	3	-	-	3
26	21BS004	Advanced Materials of Renewable Energy	1,7	3	-	-	3
27	21BS005	Applied Linear Algebra for Engineers	1,12	3	-	-	3
28	21CE019	Green Buildings	1,7,12	3	-	-	3
29	21EE017	Sustainable Energy	2,3,12	3	-	-	3
30	21ME019	Total Quality Management	1,11	3	-	-	3
31	21EC011	Communication Technologies	1,2	3	-	-	3
32	21CS020	Applications of Artificial Intelligence	2,3,6,7	3	-	-	3
33	21CH016	Green Technologies	2,6,7	3	-	-	3
34	21IT015	Human Computer Interaction	1,7	3	-	-	3
35	21BS006	Handling of Industrial waste and waste water	1,7	3	-	-	3
36	21OE001	Robotics and Automation	5,6,7	3	-	-	3
37	21OE002	Introduction to IoT	1,2	3	-	-	3
38	21OE003	Fundamentals of Image processing	1,2	3	-	-	3
39	21OE004	Fundamentals of Data Acquisition systems	1,2	3	-	-	3
40	21OE005	Airport Operations Management	2,4,11,12	3	-	-	3
41	21OE006	Fundamentals of Embedded Systems	1,2	3	-	-	3
42	21OE007	Remote Sensing and GIS	1,2,5,7,10	3	-	-	3
43	21OE008	Big Data Analytics	1,7	3	-	-	3
44	21OE009	Fundamentals of Cyber Security	3,6,8	3	-	-	3
45	21OE010	Smart Cities	7,12	3	-	-	3
46	21OE011	Nano Materials and Thin Film Technology	1,12				
47	21CSMC1	Cloud computing	2,3	3	-	-	3
48	21CSMC2	Ethical Hacking	1,2,3	3	-	-	3
49	21CSMC3	Fundamentals of Web Development	2,3,5	4	-	-	4
50	21OE012	Business Intelligence & Analytics	2,3,5	3	-	-	3
51	21OE013	Introduction To Industry 4.0 And Industrial Internet Of Things	2,3	3	-	-	3
52	21OE014	Natural Language Processing	2,3	3	-	-	3
53	21OE015	Affective Computing	1,2,3,4	3			
Audit Course							
1	21AT001	Communication Etiquette in Workplaces	-	-	-	-	-
2	21AT002	Contemporary India: Economy, Policy and Society	-	-	-	-	-
3	21AT003	Design The Thinking	-	-	-	-	-

4	21AT004	Ethics and Integrity	-	-	-	-	-
5	21AT005	Indian Heritage and Culture	-	-	-	-	-
6	21AT006	Intellectual Property Rights and Patents	-	-	-	-	-
7	21AT007	Introduction to Journalism	-	-	-	-	-
8	21AT008	Mass Media Communication	-	-	-	-	-
9	21AT009	Science, Technology and Development	-	-	-	-	-
10	21AT010	Social Responsibility	-	-	-	-	-
11	21AT011	The Art of Photography and Film Making	-	-	-	-	-
12	21AT012	Gender Equality for Sustainability	-	-	-	-	-
13	21AT013	Women in Leadership	-	-	-	-	-
14	21AT014	Introduction to Research Methodology	-	-	-	-	-
15	21AT015	Climate Change and Circular Economy	-	-	-	-	-
B. Tech. (Honors)							
Domain I (Data Engineering)							
01	21CSH11	Advanced Data Structures	1,2,3,12,PSO2	4	-	-	4
02	21CSH12	Advanced Databases	1,2,3,10,12,PSO2	4	-	-	4
03	21CSH13	Programming, Data Structures and Algorithms Using Python	1,2,3,4,5,PSO2	4	-	-	4
04	21CSH14	Bioinformatics	1,2,3,4,5,PSO2	4	-	-	4
Domain II (Modern Software Engineering)							
01	21CSH21	DevOps	1,2,3,5,8,10,PSO1, PSO2	4	-	-	4
02	21CSH22	Design Patterns	1,2,3,5,10,PSO1	4	-	-	4
03	21CSH23	Advanced Software Engineering	1,2,3,5,11, PSO1	4	-	-	4
04	21CSH24	Robotic Process Automation	1,2,3,4,5, 6,8, 10,PSO1	4	-	-	4
Domain III (Security)							
01	21CSH31	Computer Systems Security	1,2,3,5,6	4	-	-	4
02	21CSH32	Python Programming for Security	1,2,3,4,5	4	-	-	4
03	21CSH33	Management of Information Security	1,2,3,6,7,11	4	-	-	4
04	21CSH34	Computer Forensics	1,2,3,4,5,6,8,10,PSO1	4	-	-	4
Domain IV (User Interface Design)							
01	21CSH41	Computer Graphics	1,2,3,4,5,10,12,PSO1, PSO2	4	-	-	4
02	21CSH42	Multimedia Systems	1,2,3,4,5,10,12,PSO1, PSO2	4	-	-	4
03	21CSH43	Human Computer Interaction	1,2,3,4,5,6,8,10,12, PSO1,PSO2	4	-	-	4
04	21CSH44	Mobile Programming	1,2,3,4,5,6,7,8,10,12, PSO1,PSO2	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 Nano-Structures and Applications	1,3,4,5	4	-	-	4
03	21CHM23	Energy, Environmental and Biomedical Nanotechnology	1,2,3,7	4	-	-	4
04	21CHM24	Industrial Application of Nanotechnology	2,3,5,7	4	-	-	4

Environment 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
05	21CEM15	Environmental Pollution	2,3,6	4	-	-	4
Artificial Intelligence & Machine Leaning							
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	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	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 Semantic Analysis	2,4	4	-	-	4
Computer Systems Programming							
01	21CSM41	Programming Fundamentals	1,2,3	4	-	-	4
02	21CSM42	Data Structures & Algorithms	1,2,3,4	4	-	-	4
03	21CSM43	Fundamentals of Databases	1,4	4	-	-	4
04	21CSM44	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 Lab view	1,2	4	-	-	4
04	21ECM24	Medical Robotics	1,2,3	4	-	-	4
Communications and 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 Navigation 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
Smart City Management							
01	21EEM21	Fundamentals of Smart City	2,3	4	-	-	4
02	21EEM22	Smart City Infrastructure	2,3,5,6,7, 11	4	-	-	4
03	21EEM23	Computational Methods for Smart City Management	3,5	4	-	-	4
04	21EEM24	Communication Technologies and Mobility for Smart City	2,3	4	-	-	4
Industrial Applications and Control							
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 Engineering	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

5th Semester

21ECE02 Microprocessors and Microcontroller Programming

3 0 2 4

Course Outcomes

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

1. Summarize the architecture of 8086 microprocessor
2. Execute assembly language programs of 8086 microprocessor is a
3. Demonstrate the memory interfacing with 8086 microprocessor
4. Explain the architecture of 8051 microcontroller
5. Implement assembly language programs of 8051 microcontroller
6. Demonstrate the interfacing of peripherals with 8051 microcontroller

COs – POs Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	1	-	-	-	-	-	-	-	-	-	3	-
CO2	3	2	2	2	3	-	-	-	-	-	-	-	3	-
CO3	3	2	2	2	2	-	-	-	-	-	-	-	3	-
CO4	2	1	1	-	-	-	-	-	-	-	-	-	3	-
CO5	2	2	2	2	3	-	-	-	-	-	-	-	3	-
CO6	3	3	3	3	3	-	-	-	-	-	-	-	3	-

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

Unit I **14+8 Hours**

8086 Microprocessors and Assembly Language Programming

Introduction and evolution of processors, Architecture of 8086 microprocessor, Register organization of 8086, Memory segmentation, Addressing Modes.Instruction set of 8086 microprocessor: Data transfer instructions, Arithmetic instructions, Logical Instructions, String instructions, Stack related instructions, Branching instructions, Assembler directives.

Data transfer instructions of 8085 microprocessor, Architecture of 8085 microprocessor

Practical Components

1. Data transfer program using different addressing modes in assembly language programming.
2. Perform arithmetic operations on 8 bit and 16 bit numbers in assembly language programming.
3. Data transfer program using string instruction in assembly language programming.
4. Program for data conversion in assembly language programming.

Unit II **10+8 Hours**

8086 Operational Modes and Peripheral Interfacing

Procedures and macros, Stack Structure of 8086, Minimum and Maximum mode operations of 8086 with timing diagrams, Memory interfacing, 8255 Programmable Peripheral Interfacing.

Dynamic RAM, Direct memory access concept

Practical Components

1. Write assembly language program using procedure.
2. Program to arrange three bytes in ascending and descending order in assembly language.
3. Program to reject negative numbers from a series of bytes.
4. Program to reject odd numbers from a series of bytes.

Unit III **12+8 Hours**

8051 Microcontroller and Assembly Language Programming

Comparison between microprocessor and microcontroller, 8051 family microcontroller, RAM architecture of 8051, Integrated Development Environment (IDE), Pin description of 8051 microcontroller, Machine cycle. Addressing Modes, Instruction set of 8051: Data transfer instructions, Arithmetic instructions, Logical Instructions, Stack related instructions, Branching instructions. Programming and Applications of Timers, Interrupts, Universal Asynchronous Receiver Transmitter (UART).

External memory interfacing with 8051 microcontroller, various constituents of hex file

Practical Components

1. Program to perform arithmetic operations.
2. Program to toggle the LED in assembly language.
3. Programming and interfacing of traffic light logic in assembly language.
4. Program to generate square wave using interrupts.

Unit IV **12+8 Hours**

Interfacing with 8051 microcontrollers with External Peripherals

Interfacing with 8051 microcontrollers with: Keypad matrix, LCD, Seven segment displays, Stepper motor, Analog to Digital Converter (804), Digital to Analog Converter (808).

Interfacing of temperature sensor (LM 35) with 8051, interfacing of relay with 8051

Practical Components

1. Programming and interfacing of the key pad matrix.
2. Programming and interfacing of 7-segment display to display numbers from 0 to 9
3. Programming and interfacing of 7-segment display to display possible characters/symbols
4. Programming and interfacing of the LCD.

Total: 48+32 Hours

Textbook (s)

1. A.K. Ray & K. M Bhurchandi, Advanced Microprocessors & peripherals, Tata McGraw-Hill, 3rd Edition, 2012
2. Muhammad Ali Mazidi, Janice GillispieMazidi and Rolin D. McKinlay, The 8051 Micro controller and Embedded systems: using assembles and C, Pearson, 2nd Edition,2007

Reference (s)

1. D.V.Hall, Microprocessor and Interfacing, Tata McGraw Hill Publishing Company, 2nd Edition 2006
2. N. Sentil Kumar, M Sarvanan, S Jeevananthan, Microprocessors and Microcontrollers, Oxford University Press, 1st Edition, 2010
3. Kenneth J Ayala, The 8051 Microcontroller Architecture, Programming and Applications, Thomson Publishers, 3rd Edition, 2004

21CS502 Artificial Intelligence and Machine Learning

3 0 0 3

Course Outcomes

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

1. Illustrate the scope of Artificial Intelligence in the real world
2. Demonstrate various machine learning algorithms and its preliminaries
3. Summarize and learn various supervised learning algorithms
4. Model the concepts of classification and regression
5. Summarize and learn various unsupervised learning algorithms
6. Demonstrate and resolve complex clusters using dimensionality reduction

CO-PO Mapping:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	2	-	-	-	-	-	-	-	-	2	3
CO2	3	3	3	3	-	-	-	-	-	-	-	2	3	2
CO3	2	2	2	2	-	-	-	-	-	-	-	2	2	1
CO4	3	3	3	3	-	-	-	-	-	-	-	2	3	2
CO5	2	2	2	2	-	-	-	-	-	-	-	2	2	1
CO6	3	3	3	3	-	-	-	-	-	-	-	2	3	3

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

Unit -I **12 Hours**

Artificial Intelligence Foundation to AI, Problem Solving- State Space Search, Constraint Satisfaction Problems, AI Problems, AI Types and Agents, AI Tasks, AI Techniques, Heuristic Search Techniques- Best First Search Techniques, Expert Systems – Rule based system, Model & Hybrid Based System, Reasoning in uncertain environment, Math for Machine Learning- Linear Algebra, Multi-Variable Calculus and Vectors

AI Models, AI Bots, Applications of AI, Applied AI

Unit –II **12 Hours**

Machine Learning Foundation for Machine Learning – Types of Machine Learning, Machine Learning Process, Preliminaries- Over fitting, Training, Testing and Validating Sets, Splitting and Feature Scaling, Data Pre-processing- Importing Libraries, Importing Datasets, Missing Data and Dependent Variables, Machine Learning Applications.

Rules of Probability, Bayes Theorem, Bias, Variance and Co-variance.

Unit -III **12 Hours**

Supervised Learning Classification- Naïve Bayes, Support Vector Machines, Extension to SVM, K-Nearest Neighbours, Decision Tree, Classify Vs Predict, Regression – Linear Regression, Gradient Descent and Logistic Regression, Regularization, LASSO & Ridge Regression Algorithms, Performance metrics.

Margins and Vectors, Continuous and Discrete variables

Unit -IV: **12 Hours**

Unsupervised Learning Clustering- Iterative Distance based Clustering, Association Rule Learning, K-Means Clustering and Hierarchical Clustering, Measure of Quality Clustering. Dimensionality Reduction – Feature Selection, Principal Component Analysis, Independent Component Analysis and Linear Discriminant Analysis.

Similarity and Distance Measures, Gaussian and Normal Deviation

Total: 48 Hours

Text Book (s)

1. Russel and Norvig, Artificial Intelligence A modern Approach, 4th Edition, Pearson Education 2021
2. Raschka, Sebastian and Mirjalili, Vahid, Python Machine Learning, 3rd Edition, Packt Publishing., 2020
3. Stephen Marsland- Machine Learning – An Algorithmic Perspective – Second Edition – Chap Man & Hall CRC Press, 2015

Reference (s)

1. E. Rich K. Knight and B.Nair – Artificial Intelligence– Third Edition – Tata McGraw Hill, 2017
2. Tom M. Mitchell, ―Machine Learning, McGraw-Hill Education (India) Private Limited, 2013.

21CS503 Computer Networks

3 0 2 4

Course Outcomes

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

1. Illustrate the knowledge of the basic Data Communication System and Computer Network Systems.
2. Classify various wired and wireless transmission media for data communication networks
3. Demonstrate different techniques of error detection and correction to detect and solve error bit during data transmission
4. Illustrate the subnetting, routing mechanisms and congestion issues in network design.
5. Experiment with different network tools.
6. Summarize the internal functionalities of main protocols such as HTTP, SNMP, TCP, UDP, IP

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	-	2	1	-	2	-	-	-	-	2	1	-
CO2	3	2	-	2	1	-	2	-	-	-	-	2	2	-
CO3	2	3	-	2	2	-	2	-	-	-	-	2	1	-
CO4	3	3	-	3	3	-	2	-	-	-	-	2	1	-
CO5	2	3	-	3	3	-	2	-	-	-	-	2	2	-
CO6	3	1	-	3	2	-	2	-	-	-	-	2	2	-

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

UNIT – I **12 + 8 hours**

Introduction to Data Communications: Components, Data Representation, Data flow

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet..

Physical Layer: Transmission media: Guided media- twisted pairs, coaxial cable, fiberoptics, unguided media-Wireless transmission, Switching Techniques: Circuit Switching-Packet Switching-Message Switching.

Taxonomy of networking devices

Practical Components:

1. Familiarization with Networking Components and devices: LAN Adapters - Hubs – Switches - Routers etc.
2. Familiarization with Transmission media and Tools: Co-axial cable - UTP Cable - Crimping Tool - Connectors etc.
3. Preparing the UTP cable for cross and direct connections using crimping tool.
4. a) To configure the IP address for a computer connected to LAN and to configure network parameters of a web browser for the same computer.
   b) Configuration of TCP/IP Protocols in Windows

UNIT – II **12 + 6 hours**

Design Issues: Framing-error detection and correction-CRC

Elementary Data link Protocols: Stop and wait-Sliding, Window protocols: Go-back-n-Selective Repeat, **Medium Access sub layer**: Channel allocation methods, Multiple Access protocols: ALOHA-CSMA-IEEE Standard 802.3 and Ethernet

Data Link Control Protocols: HDLC-SLIP-PPP

Practical Components:

1. Implement the data link layer framing methods:
   a) Character stuffing method
   b) Bit Stuffing method
2. Implement on a data set of characters the two CRC polynomials: CRC 12 - CRC 16
3. Develop a simple data link layer that performs the flow control using
   a) The sliding window protocol,

b) Go-Back-N Protocol.

UNIT – III **12 + 8 hours**

NetworkLayer:Designissues,Routingalgorithms:shortestpathrouting,distance vector routing, Flooding, Hierarchicalrouting, Broadcast, Multicast, Congestion Control Algorithms - Approaches to Congestion Control, Quality of Service-leaky bucket algorithm, token bucket algorithm, The Networklayerintheinternet-IPv4 Protocol, IP Addresses, Subnetting.

Internet control protocols

Practical Components:

1. Implement Dijkstra's algorithm to compute the Shortest path through a graph.
2. Take an example subnet graph with weights indicating delay between nodes. Now Obtain Routing table art each node using distance vector routing algorithm.
3. Take an example subnet of hosts. Obtain broadcast tree for it
4. Write a program for congestion control using Leaky bucket algorithm.

UNIT – IV **12 + 10 hours**

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCPandUDPprotocols. Application Layer –Domain name system, SNMP, Electronic Mail, World Wide Web, HTTP

Proxy Servers, Data compression

Practical Components:

1. a) Installing of internal modem and connecting to Internet.
   b) To configure WiFi for your PC.
2. Wireshark
   a) Packet Capture Using Wire shark
   b) Starting Wire shark
   c) Viewing Captured Traffic
   d) Analysis and Statistics & Filters.
3. Do the following using NS3 Simulator
   a) NS3 Simulator-Introduction and installation
   b) Simulate to Find the Number of Packets Dropped
   c) Simulate to Find the Number of Packets Dropped by TCP/UDP
   d) Simulate to Find the Number of Packets Dropped due to Congestion
   e) Simulate to Compare Data Rate& Throughput.
   f) Simulate to Plot Congestion for Different Source/Destination
   g) Simulate to Determine the Performance with respect to Transmission of Packets
4. Write a program for how to connect and transfer data between two nodes with each other.
5. Study and build a sample network topology to configure it for dynamic routing protocol using NS3

Total: 48+32 hours

Textbook (s)

1. Andrew S Tanenbaum, Computer Networks, 5th Edition, Pearson Education /PHI, 2013
2. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, Tata McGraw Hill Higher Education, 2013
3. Reference (s)
4. Willam Stallings, Data and Computer Communications, 8th Edition, Pearson Prentice Hall, 2007
5. W.A. Shay, Thomson, Understanding communications and Networks, 3rd Edition, Cengage Learning, 2005

21CS504 Theory of Computation

3 0 0 3

Course Outcomes

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

1. Define State machines, languages and computations
2. Design a state machine with and without outputs
3. Compare regular grammars and regular languages
4. Relate Context free languages and Context free grammars
5. Design Pushdown automata for Context free languages
6. Illustrate decidable and un-decidable problems

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	-	-	-	-	-	-	-	-	2	3	-
CO2	3	2	2	-	-	-	-	-	-	-	-	2	3	-
CO3	3	2	3	-	-	-	-	-	-	-	-	2	3	-
CO4	3	3	2	-	-	-	-	-	-	-	-	2	2	-
CO5	3	2	3	-	-	-	-	-	-	-	-	2	2	-
CO6	3	3	2	-	-	-	-	-	-	-	-	2	2	-

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

Unit I **13 Hours**

Finite Automata: Introduction to formal languages- language operations- Finite automata model-Deterministic Finite Automata – Nondeterministic Finite automata - Recognition of a language by an Automaton - Equivalence of DFA and NFA - Finite Automata with Null-Closure - Minimization of Finite Automata - Equivalence of FAs - Mealy and Moore Machine, Equivalence of Mealy and Moore machines.

Mathematical induction, strong principle, Complementation of Finite automata, Compound Automata

Unit II **11 Hours**

Regular Languages: Regular Sets-Regular expressions- Equivalence of Finite automata and regular expressions –Pumping Lemma for Regular Languages - Closure Properties of Regular sets. Regular Grammars: Right linear grammar- Left linear grammar- Equivalence of RLG and finite automata-Equivalence of LLG and finite automata.

Myhill-Nerode theorem –applications of regular expressions

Unit III **13 Hours**

Context free language- Closure Properties of Context free languages, pumping lemma for context free languages- Context-free grammar-simplification of Context free Grammar-Normal forms: Chomsky Normal form and Greibach Normal form.

Pushdown Automata: PDA Model-Design-Acceptance by empty stack and final state- Non-deterministic PDA. Context free grammar to Pushdown automata conversion.

Context Sensitive Languages, Linear bounded automata

Unit IV **11 Hours**

Turing Machine: Chomsky classification of languages- Turing Machine Model - Design of TM- Turing Machines variants–Undecidable languages - properties of Recursive and Recursively Enumerable Languages -Post Correspondence Problem - Halting Problem - Polynomial Reducibility, P – NP, NP – Complete and NP – Hard problems

Computable Functions - Constant Functions, Recursive Functions

Total 48 Hours

Text Books:

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

Reference books:

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.

21CS004 Principles of Programming Languages

3 0 0 3

Course Outcomes

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

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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	2	2	-	-	-	-	-	-	-	2	-	-
CO2	3	2	1	2	-	-	-	-	-	-	-	2	-	-
CO3	3	1	3	1	-	-	-	-	-	-	-	2	-	-
CO4	3	2	2	2	-	-	-	-	-	-	-	2	-	-
CO5	3	2	2	3	-	-	-	-	-	-	-	2	-	-
CO6	3	2	2	3	-	-	-	-	-	-	-	2	-	-

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

21CS005 Mobile Computing

3 0 0 3

Course Outcomes

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

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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	3	-	-	-	2	-	-	1	-	-	-	-
CO2	2	2	2	-	-	-	2	-	-	1	-	-	-	-
CO3	2	2	1	-	-	-	2	-	-	1	-	-	-	-
CO4	2	2	2	-	-	-	2	-	-	1	-	-	-	-
CO5	2	2	2	-	-	-	2	-	-	1	-	-	-	-
CO6	2	2	2	-	-	-	2	-	-	1	-	-	-	-

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: Comunication Modes, Basic Network Designs, Cellular Infrastrures

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: Charactestics, 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)

21CS006 Distributed Operating Systems

3 0 0 3

Course Outcomes

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

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's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO2	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO3	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO4	2	2	2	-	-	-	-	-	-	-	-	1	1	-
CO5	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO6	3	1	2	-	-	-	-	-	-	-	-	1	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 FileSystems **12 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: 48Hours

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

21CS507 Artificial Intelligence and Machine Learning Lab

0 0 3 1.5

Course Outcomes

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

1. Construct rule-based systems for solving real-world problems.
2. Demonstrate the function of data repositories and pre-processing tools.
3. Apply and evaluate classification techniques for solving classification problems.
4. Build regression models to accurately predict outcomes.
5. Apply clustering techniques to group data and analyze the outcomes.
6. Simplify complex data using dimensionality reduction.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	3	2	-	-	-	-	-	-	3	3	3
CO2	2	-	-	2	3	-	-	-	-	-	-	2	3	3
CO3	3	2	-	3	3	-	-	-	-	-	-	2	3	3
CO4	2	2	-	2	3	-	-	-	-	-	-	2	2	2
CO5	3	2	-	3	2	-	-	-	-	-	-	2	2	2
CO6	3	2	-	3	3	-	-	-	-	-	-	2	2	2

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

List of Experiments:

Expert System

1. Develop a Rule Based system (Consider any Problem from the Annexure any Real Time)

Data Pre-Processing and Data Pre-Processing Tools

2. Importing Datasets, Missing Data, Feature Encoding, Handling Outliers
3. Splitting Datasets to Training Dataset and Test Dataset, Feature Scaling, Data Sampling

Classification

4. K-Nearest Neighbor(K-NN) Classifier
5. Support Vector Machine Classification
6. Naïve Bayes Classifier
7. Decision Tree Classification

Regression

8. Simple Linear Regression
9. Multiple Linear Regressions
10. Polynomial Regression
11. Logistic Regression

Clustering

12. Association Rule Learning
13. K Means Clustering
14. Hierarchical Clustering

Dimensionality Reduction

15. Principal Component Analysis
16. Linear Discriminate Analysis

Text Book (s)

1. Raschka, Sebastian and Mirjalili, Vahid, Python Machine Learning, 3rd Edition, Packt Publishing., 2020
2. Stephen Marsland- Machine Learning – An Algorithmic Perspective – Second Edition – Chap Man & Hall CRC Press, 2015

Reference (s)

1. Tom M. Mitchell, ―Machine Learning, McGraw-Hill Education (India) Private Limited, 2013.
2. Machine Learning A-Z: Hands-On Python, Udemy.

Annexure:

1. Banking System
2. Weather System
3. Medical System
4. Gaming System
5. Dressing System
6. Disease Diagnose or Prediction System

Augmented Experiments:

1. Rule Based Expert System: Design, Suiting and Shirting for an event/ occasion
2. Gaming: Chess, Water Jug Problem, 8 Queens Problem
3. Real Time: Traffic Lights
4. Real Time: Banking Transactions (Only Debit and Credit)
5. Mathematical Approach for Theorem Proving
6. Financial Approach for Loans via Cash / Vehicle / Gold
7. Medical Expert Systems
8. Disease Classification Strategies
9. Disease Prediction Strategies
10. Work on a Real Dataset.

21TPX01 Term Paper

0 0 3 1.5

Course Outcomes

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

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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	-	3	2	3	-	3	2	3	-	3	2	2
CO2	2	2	-	3	3	3	-	2	3	3	-	3	2	2
CO3	3	3	-	3	2	3	-	3	2	3	-	3	2	2
CO4	3	3	-	3	2	3	-	3	3	3	-	3	2	2
CO5	2	2	-	3	3	3	-	2	2	3	-	3	2	2
CO6	3	2	-	3	1	3	-	3	3	3	-	3	2	2

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

Term Paper: The Term Paper is a self-study report and shall be carried out either during 5th or 6th semester in choice with Mini Project. Every student will take up this term paper individually 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 as nominated by HoD with the approval of CoE

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

21SIX01 Summer Internship I

0 0 0 1

Course Outcomes

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

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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	2	1	2	3	1	3	3	3	3	3	3	3
CO2	3	3	3	3	2	2	1	3	3	1	1	3	3	1
CO3	3	2	3	2	2	1	1	3	2	3	3	2	3	3
CO4	3	3	3	3	3	3	2	3	3	3	3	3	3	3
CO5	3	3	3	3	3	3	3	3	3	3	3	3	3	3
CO6	3	3	3	3	3	3	3	2	3	3	3	3	3	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.

6th Semester

21CS601 Compiler Design

3 0 0 3

Course Outcomes

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

1. Explain different translators and acquire knowledge of compiler & its Phases.
2. Identify formal grammars for specifying the syntax and Semantics of programming languages.
3. Construct parse table for a given grammar.
4. Examine the intermediate code for a given program.
5. Illustrate the symbol table management and storage organization.
6. Develop code optimization techniques to improve the performance of a program.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	-	-	-	-	-	-	-	-	-	2	2	-
CO2	3	2	3	-	-	-	-	-	-	-	-	2	2	-
CO3	3	3	3	-	-	-	-	-	-	-	-	1	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-	1	1	-
CO5	3	2	-	-	-	-	-	-	-	-	-	1	1	-
CO6	3	2	3	-	-	-	-	-	-	-	-	2	2	-

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

Unit I **12 Hours**

Overview of Language Processing-Compiler-Assembler-Interpreters-Linkers & Loaders-Phases of a Compiler-Lexical Analysis-Role of Lexical Analysis-Token-Patterns and Lexemes-Lexical Errors-Regular definitions for the Language constructs- Transition diagrams for recognition of Tokens-Reserved Words and Identifiers.

Input buffering, LEX tool

Unit II **12Hours**

Syntax Analysis-Top-down Parsing-First and Follow-Recursive predictive parsing- Non-Recursive Predictive Parsing-LL(1) Grammar-Bottom-upparsing-ShiftReduceParsing-ModelofanLRParsers-ConstructionofSLR(1)Tables-Constructionof CLR(1)-LALR(1)-Parsing tables

Handling Ambiguous grammars- Dangling ELSE ambiguity

Unit III **12 Hours**

Semantic Analysis - SDT: Synthesized and Inherited Attributes-IntermediateCode-ThreeAddressCode-Quadruples-Triples-Indirect Triples-Abstract Syntax Trees-DAG for Expressions-Symbol Table management-RuntimeEnvironment:StorageOrganization-StorageAllocationstrategies

Data structures used in symbol table, Three address code for control statements

Unit IV **12 Hours**

MachineIndependentCodeOptimization-CommonSub-expressionElimination-ConstantFolding-CopyPropagation-Dead Code Elimination-Strength Reduction-Loop Optimization-Basic Blocks-Flow Graph-DAGforbasicBlocks-MachineDependentCodeOptimization:PeepholeOptimization-RegisterAllocation-InstructionScheduling

Code generation algorithm – optimization among basic blocks

Total: 48 Hours

Text Books:-

1. AlfredVAho,MonicaSLam,RaviSethi,JeffreyD.Ullman,Compilers,PrinciplesTechniquesandTools,2ndEdition,Pearson,2007.
2. V.Raghavan,PrinciplesofCompilerDesign,2ndEdition,TMH,2011.

Reference Books:-

1. NandiniPrasad,PrinciplesofCompilerDesign,2ndEdition,Elsevier,2012.
2. YunlinSuSongY.Yan,Principles of Compilers,ANewapproachtoCompilersincludingtheAlgebraicMethods,Springer,2011.

21CS602 Internet of Things

3 0 0 3

Course Outcomes

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

1. Illustrate IoT framework, architecture and design principles of M2M Communication
2. Outline the design principles of Web and Internet Connectivity and its protocols
3. Summarize MAC and IP addressing in IoT, 6LowPAN and LoRaWAN protocols
4. Identify the suitable sensor technology for data collection and how to store and use it for Computing
5. Develop business models in IoT by identifying vulnerabilities and attacks involving in IoT
6. Inspect an application using IoT technology

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	2	3	-	-	3	-	-	-	-	-	2	-
CO2	3	-	2	2	-	-	2	-	-	-	-	-	3	-
CO3	3	-	1	1	-	-	2	-	-	-	-	-	2	-
CO4	3	2	3	2	-	-	2	-	-	-	-	-	3	-
CO5	3	2	3	2	-	-	3	-	-	-	-	-	3	-
CO6	3	2	3	3	-	-	3	-	-	-	-	-	3	-

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

UNIT I **12Hours**

Overview of IoT: Overview of Wireless Sensor Networks, Overview of Internet of Things, IoT Conceptual Framework, IoT Architectural View, Technology Behind IoT, Sources of IoT, M2M Communication.
Design Principles: IoT/M2M Systems Layers and Design Standardization, Communication Technologies, Data Enrichment, Data Consolidation and Device Management at Gateway

Examples of IoT, Ease of Designing and Affordability

UNIT II **12 Hours**

Design Principles for Web Connectivity: Introduction, Web Communication Protocols: Constrained Applications Protocol (CoAP), Lightweight Machine-to-Machine Communication**;** Message Communication Protocols: Message Queue Telemetry Transport (MQTT);
Internet Connectivity Principles: Introduction, Internet Connectivity, Internet-Based Communication, IP Addressing in the IoT, Media Access Control, Introduction to 6LowPAN and LoRaWAN

***Application Layer Protocols**: HTTP, HTTPS, FTP and Telnet*

UNIT III **12 Hours**

Sensor Technology: Sensing the Real World using Analog and Digital Sensors, Industrial IoT, Automotive IoT, Actuator, RFID Technology – Principles, Architecture, Applications & Components, Web of Things of

RFIDs. **Data Collection, Storage and Computing:** Introduction, Cloud computing Paradigm for Data Collection, Storage and Computing, Everything as a Service and Cloud Service Models for IoT

Sensor Data Communication Protocol: *Serial Bus – USB, CAN*

UNIT IV **12 Hours**

Business Models and Processes using IoT: Introduction, Business Models, Business Scenarios in IoT. **IoT Privacy, Security and Vulnerabilities Solutions:** Security and Privacy Requirements, Threat Analysis, IoT Layered Attacker Model, Access Control and Secure Message Communication, Security Models

Case Studies: Smart Home, Smart City, Precision Agriculture ***IoT Hardware:*** *Raspberry pi, Arduino.*

Textbook:

1. Raj Kamal, "Internet of Things: Architecture and Design Principles". TMH Publications, 2017.
2. Vijay Madisetti and ArshdeepBahga, "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014

Reference:

1. Ovidiu Vermesan& Peter Friess, "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems", River Publishers Series in Communications, 2017.

21CS603 Software Engineering

3 0 0 3

Course Outcomes

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

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

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	2	-	-	-	-	-	2	-	3	-
CO2	3	2	3	-	2	-	-	-	-	-	3	-	3	-
CO3	3	2	3	-	3	-	-	-	-	-	3	-	3	-
CO4	3	2	3	-	2	-	-	-	-	-	3	-	3	-
CO5	3	2	3	-	2	-	-	-	-	-	3	-	3	-
CO6	3	2	2	-	2	-	-	-	-	-	3	-	3	-

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, Sattic 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 InternationalPublication,2008

21CS007 Cloud Computing Essentials

3 0 2 4

Course Outcomes

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

1. Interpret the architecture and infrastructure models of cloud computing, strengths, and limitations of cloud computing.
2. Explain 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. Discuss 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's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	-	3	-	-	3	2	3	1	-	-	-	-	-	-
CO2	-	3	-	-	1	1	2	3	-	-	-	-	-	-
CO3	-	2	-	-	2	3	1	2	-	-	-	-	-	-
CO4	-	2	-	-	1	3	3	1	-	-	-	-	-	-
CO5	-	2	-	-	1	3	1	2	-	-	-	-	-	-
CO6	-	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)

1. Michael Miller, Cloud Computing-Web Based Applications that change the way you work and collaborate online, 1st Edition, Pearson Education, Publishing, 2011
2. 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
3. David S Linthicum, Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide, 1st Edition, Addison-Wesley, 2009
4. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, SubraKumaraswamy, ShahedLatif, O’Reilly, SPD, rp2011.

21CS008 Cryptography and Network Security

3 0 2 4

Course Outcomes

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

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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	-	3	2	-	-	-	-	-	-	2	-
CO2	2	2	2	-	3	2	-	-	-	-	-	-	2	-
CO3	2	2	2	-	3	1	-	-	-	-	-	-	2	-
CO4	2	2	2	-	2	1	-	-	-	-	-	-	2	-
CO5	2	2	2	-	3	2	-	-	-	-	-	-	2	-
CO6	2	2	2	-	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

21CS606 Case Tools Lab

0 0 3 1.5

Course Outcomes

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

1. Apply basics of Unified Modeling Language (UML) to illustrate its importance in software engineering and system design
2. Design various use cases to showcase the functional requirements of the system.
3. Construct UML diagrams for static and dynamic views of the system
4. Inference various case studies and design the model
5. Construct the code from the model for the given application
6. Demonstrate and deploy the solutions for various real time problems.

COs–POs Mapping

CO’s	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	-	2	3	1	-	-	-	-	2	2	-	2	-
CO2	2	3	2	2	2	-	-	-	-	3	2	-	2	-
CO3	2	3	3	3	3	-	-	-	-	3	2	-	2	-
CO4	2	-	3	2	3	-	-	-	-	3	2	-	3	-

CO5	2	-	2	2	3	-	-	-	-	2	2	-	3	-
CO6	2	3	2	2	2	-	-	-	-	-	2	-	3	-

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

List of Experiments:

Case studies given below should be Modeled using Rational Rose tool in different views i.e. Use case diagram, Class diagram, Sequence Diagram, Collaboration Diagram, State Diagram, Activity Diagram, Component Diagram, Deployment Diagram and Forward and Reverse Engineering.

CASE STUDY 1:

LIBRARY MANAGEMENT SYSTEM:

This case study on the library management system gives us the complete information about the library and the daily transactions done in a Library. We need to maintain the record of new s and retrieve the details of books available in the library which mainly focuses on basic operations in a library like adding new member, new books, and up new information, searching books and members and facility to borrow and return books.

CASE STUDY 2:

A HOSPITAL MANAGEMENT SYSTEM:

Entry in the hospital management system is done through system users. Master form for users is used to create user profile. Whole system is display in user's selected language. For example if user knows Germany then whole hospital management system is display in German language after login. Above all modules are related to patient. So when patient is admit then patient information is enter from patient entry form like patient name, patient initial, patient sex, patient birth date, blood group, patient ID.

CASE STUDY 3:

AUTOMATED TELLER MACHINE (ATM)

Software is designed for supporting a computerized ATM banking network. All the process involved in the bank is computerized these days. All the accounts maintained in the bank and also the transactions effected, including ATM transactions are to be processed by the computers in the bank. An ATM accepts a relevant cash card, interacts with user, communicates with the central system to carry out the transaction, dispenses cash, and prints receipts. The system to be designed and implemented must include appropriate record keeping and security provisions. The system must handle concurrent access to the same account.

CASE STUDY 4:

DESIGN A STUDENT REGISTRATION SYSTEM :

Each student has access to his or her course and grade information only and must be authenticated prior to viewing or updating the information. A course instructor will use the system to view the list of courses he or she is assigned for a given semester or has taught previously, view the list of students registered for the course(s) he or she is teaching and record final grades for each student in the course(s). TA assignments will also be viewable through this system. Instructors must also be authenticated prior to viewing or updating any information.

CASE STUDY 5:

MULTITHREADED AIRPORT SIMULATION:

Everyday number of airplanes lands and takes off from airport. It is the responsibility of the Air Traffic Control (ATC) to regulate these planes. The aim of this simulation is to reconstruct the events occurring during landing or take off. Whenever plane enters the RADAR space, RADAR signals the ATC about the plane. Then the pilot sends the plane details. ATC checks the runway and decides priority. ATC signals the pilot whose plane is having highest priority to land/take off. Then the pilot performs the corresponding command.

CASE STUDY 6:

POINT OF SALE:

Shops are the integral part of any civilization. They are present in every culture since ages. This application emphasizes on the transaction going on between the customer and shop keeper during sale of product. The customer approaches the shop keeper/ sales boy and places his order. The shop keeper forwards the order to sales boy who fetches the required goods from inventory. Then the shop keeper calculates bill and issues to the customer. The customer on paying the bill takes goods from the shop keeper. In turn, shop keeper fetches goods from supplier.

List of Augmented Experiments

1. Create a UML model for Passport automation system.
2. Create a UML model for Book bank
3. Create a UML model for Exam Registration
4. Create a UML model for Stock maintenance system.
5. Create a UML model for Online course reservation system
6. Create a UML model for E-ticketing
7. Create a UML model for Credit card processing
8. Create a UML model for e-book management system
9. Create a UML model for Recruitment system
10. Create a UML model for Quiz system

Reading Material (s)

1. Case tools Lab manual, Department of CSE, GMRIT, Rajam

21MPX01 Mini Project

0 0 3 1.5

Course Outcomes

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

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	PSO1	PSO2
CO1	3	3	2	3	2	3	2	2	2	2	3	3	3	3
CO2	3	3	3	2	3	3	3	3	2	3	3	3	3	3
CO3	2	3	3	3	3	3	3	2	2	2	3	3	3	3
CO4	2	2	3	3	3	3	3	3	3	3	3	3	3	3
CO5	3	3	2	2	3	3	3	3	3	2	3	3	3	3
CO6	3	3	2	2	3	3	3	3	3	3	3	3	3	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

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

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Develop life skills with behavioral etiquettes and personal grooming.
3. Assess analytical and aptitude skills.
4. Develop algorithms for engineering applications.
5. Solve engineering problems using software.
6. Utilize simulation tools for testing

COs –POs Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	1	1	1	1	-	-	-	2	3	3	3	-	-
CO2	1	1	3	1	1	-	-	-	2	3	3	3	-	-
CO3	1	1	3	1	1	-	-	-	2	2	3	3	-	-
CO4	2	2	3	2	2	-	-	-	3	3	2	3	-	-
CO5	3	2	3	2	2	-	-	-	3	2	2	3	-	-
CO6	2	2	2	2	3	-	-	-	3	2	3	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: (Advanced Competitive Programming) **16 Hours**

Module-1:
Non-Linear Data Structures: Heaps - Maps- Trees- Graphs - Tries.

Week	Topic	Practice Problem Links
1	Searching	https://www.hackerearth.com/practice/algorithms/searching/linear-search/practice-problems/
		https://www.codechef.com/practice/topics/binary-search
		https://leetcode.com/tag/binary-search/
		https://www.interviewbit.com/practice/#topics[]=binary-search
2	Sorting	https://www.codechef.com/practice/topics/sorting
		https://leetcode.com/tag/sorting/
		https://leetcode.com/tag/merge-sort/
		https://www.codechef.com/practice/topics/sorting
3	Heaps & Maps	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=heap
		https://www.interviewbit.com/courses/programming/heaps-and-maps
		https://workat.tech/problem-solving/topics/bst-heaps-and-map/practice
4	Trees - I	https://www.codechef.com/practice-old/tags/trees
		https://leetcode.com/problemset/?page=1&topicSlugs=binary-tree
		https://www.hackerearth.com/practice/data-structures/trees/binary-and-nary-trees/practice-problems/
		https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=trees
5	Trees - II	https://www.codechef.com/practice-old/tags/tree-data-structure
		https://leetcode.com/problemset/?page=1&topicSlugs=binary-search-tree
6	Trees - III	https://leetcode.com/problemset/?page=1&topicSlugs=segment-tree
		https://www.interviewbit.com/courses/programming/tree-data-structure
7	Graphs - I	https://www.codechef.com/practice-old/tags/graphs
		https://www.hackerrank.com/domains/algorithms?filters%5Bsubdomains%5D%5B%5D=graph-theory
8	Graphs - II	https://leetcode.com/problemset/?page=1&topicSlugs=graph
		https://www.interviewbit.com/courses/programming/graph-data-structure-algorithms

Module-2:
Algorithm design Techniques: Backtracking – Dynamic Programming – Greedy Approach.

9	Backtracking - I	https://www.codechef.com/practice-old/tags/backtracking
		https://leetcode.com/tag/backtracking/
10	Backtracking - II	https://www.interviewbit.com/courses/programming/backtracking#problems
		https://workat.tech/problem-solving/topics/backtracking/practice
11	Dynamic Programming - I	https://www.codechef.com/practice-old/tags/dynamic-programming
		https://www.hackerrank.com/domains/algorithms?filters%5Bsubdomains%5D%5B%5D=dynamic-programming
		https://www.hackerearth.com/practice/algorithms/dynamic-programming/introduction-to-dynamic-programming-1/practice-problems/
12	Dynamic Programming - II	https://leetcode.com/problemset/?page=1&topicSlugs=dynamic-programming
		https://www.interviewbit.com/courses/programming/dynamic-

		programming#problems
13	Greedy Algorithms - I	https://www.codechef.com/practice-old/tags/greedy
		https://www.hackerrank.com/domains/algorithms?filters%5Bsubdomains%5D%5B%5D=greedy
		https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/practice-problems/
14	Greedy Algorithms - II	https://www.interviewbit.com/courses/programming/greedy-algorithm#problems
		https://leetcode.com/problemset/?page=1&topicSlugs=greedy
15	Tries - I	https://www.codechef.com/practice-old/tags/tries
		https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=trie
		https://www.hackerearth.com/practice/data-structures/advanced-data-structures/trie-keyword-tree/practice-problems/
16	Tries - II	https://leetcode.com/problemset/?page=1&topicSlugs=trie
		https://workat.tech/problem-solving/topics/string-and-tries/practice

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 1

Course Outcomes

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

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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	-	-	-	-	-	-	-	-	2	3	-	3	-	-
CO2	-	-	-	-	-	-	-	-	2	3	-	2	-	-
CO3	-	-	-	-	-	-	-	-	3	3	-	2	-	-
CO4	-	-	-	-	-	-	-	-	3	2	-	2	-	-
CO5	-	-	-	-	-	-	-	-	3	2	-	2	-	-
CO6	-	-	-	-	-	-	-	-	3	2	-	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 be taken into account for CGPA calculation.

Career Path I (Artificial Intelligence& Machine Learning)
21CSC11 Exploratory Data Analytics

3 0 0 3

Course Outcomes

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

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's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	-	2	-	-	-	-	-	-	-	-	-	3
CO2	3	3	-	3	-	-	-	-	-	2	-	2	-	2
CO3	3	2	-	2	-	-	-	-	-	-	-	2	-	1
CO4	2	3	-	3	-	-	-	-	-	2	-	-	-	2
CO5	3	3	-	3	-	-	-	-	-	-	-	2	-	1
CO6	3	3	-	2	-	-	-	-	-	-	-	-	-	1

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

Unit I

Exploratory Data Analysis Fundamentals **12Hours**

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, Measurement scales-Nominal, Ordinal, Interval, Ratio. Comparing EDA with classical and Bayesian analysis, getting started with EDA: Numpy, Pandas,Scipy, and Matplotlib.

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

Objectives of Exploratory data Analysis, The applications of EDA.

Unit II **12Hours**

Visual aids for EDA

Line chart, Bar charts, Boxplot, residual plot, Scatter plot-bubble chart, scatterplot using seaborn, Area plot, stacked plot, and stem-and-leaf plot, pie chart, table chart, polar chart, histogram, lollipop chart, choosing the best chart.

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.

Descriptive statistics: Understanding statistics, distribution function (uniform, normal, exponential, binomial), cumulative distribution function, measure of central tendenacy, measure of dispersion (standard deviation, variance, skewness, kurtosis, percentiles, quartiles)

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

Unit III **12 Hours**

Correlation and Inferences from statistical tests:

Introducing correlation, covariance, Pearson's Correlation, Spearman's Rank Correlation.

Hypothesis Testing: Testing a difference in mean, testing a correlation, chi-squared tests, errors, power.

Model development and evaluation with regression techniques.

Correlation vs causation, ANOVA

Unit IV **12 Hours**

Multivariate Analysis: overview, Factor Analysis, Cluster Analysis, Discriminant analysis, EDA Case Study

Multidimensional Scaling, MANOVA vs ANOVA

Total: 48Hours

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. Downey, Allen. Think stats: exploratory data analysis. " O'Reilly Media, Inc.", 2014.
4. 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. Wendy L. MartinezAngel R. MartinezJeffrey L. Solka: Exploratory Data Analysis with MATLAB, 2nd Edition,CRC Press,2011
4. RadhikaDatar, Harish Garg : Hands-On Exploratory Data Analysis with R , Packt,2020
5. Joseph F Hair, Barry J. Babin, Rolph E. Anderson, William C. Black: Multivariate Data Analysis Cengage, 2018

21CSC12 Deep Learning

3 0 2 4

Course Outcomes

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

1. Explain the fundamental of Artificial Neural Networks
2. Identify different data representations for Neural Networks
3. Make use of different CNN models.
4. Compare different character encoding techniques.
5. Exemplify different Deep unsupervised models
6. Examine different deep learning applications.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	1	2	-	-	-	-	-	-	-	3	-
CO2	3	2	-	1	2	-	-	-	-	-	-	-	1	-
CO3	3	2	2	2	3	-	-	-	-	-	-	2	3	-
CO4	2	3	-	1	2	-	-	-	-	-	-	-	1	-
CO5	2	3	2	2	2	-	-	-	-	-	-	2	3	-
CO6	3	3	3	3	3	-	-	-	-	-	-	2	3	-

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

Unit I **12+8 Hours**

Introduction to Neural Networks

Artificial Neural Networks: Introduction, Neuron Model, Neural Network Architecture, Learning Rules, Single Layer Perceptron, Multilayer Perceptron, adaptive resonance theory (ART), Back propagation Networks : Kohnen's self-organizing networks, Hopfield network, Applications of NN, Data representations for neural networks : 0D tensors, 1D tensors, 2D tensors, tensor attributes, data tensors (Vector data, Time series data, images, video), tensor operations.

Model Parameters vs Hyperparameters, Types of activation functions, Gradient descent ,delta rule

Practical Component:

1. Build a Neural Network with back propagation algorithm and test the same using appropriate dataset.
2. The implementation of tensors and various operations on tensors.
3. Implementation of single layer perceptron and multi-layer perceptron model
4. Build a model using time series data with suitable observation

Unit II **12+10 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-task learning, end-to-end deep learning.
CNN models: AlexNet, VGG –16, Residual Networks, YOLO

Keras ,tensorflow, data augmentation ,Batch Normalization, Dropout

Practical Component:

1. Digit and Character recognizer using CNN
2. Explore AlexNet with suitable dataset
3. Explore VGG16 with suitable dataset
4. Explore ResNet with suitable dataset
5. Explore YOLO with suitable dataset

Unit III **11+8 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, Deep Boltzmann Machines

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

Practical Component:

1. One-hot encoding of words and characters using word embeddings.
2. Study the construction and working of Recurrent Neural Network
3. Explore LSTM to predict stock prices based on historic data.
4. Explore GRU to predict stock prices based on historic data.

Unit IV **12 + 6 Hours**

Applications of Deep Learning

Image segmentation, object detection, automatic image captioning, Image generation with Generative adversarial networks, video to text with LSTM models. Attention models for computer vision tasks

Batch Gradient Descent vs Stochastic Gradient Descent, limitations of deep learning

Practical Component:

1. Explore applications like Image segmenation and object detection.
2. Explore attention models for computer vision tasks with suitable example
3. Automatic image captioning and video to text models.

Total: 48+32 Hours

Textbook (s)

1. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep Learning." An MIT Press book in preparation. (2015).
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. FranchoisChollet, 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 GianmarioSpacagna, Packt Publishing; 2/e (January 16, 2020)
4. Bishop, C., M., Pattern Recognition and Machine Learning, Springer, 2006
5. Kevin P. Murphy.,MachineLearning:A Probabilistic Perspective

Career Path II (Full Stack Developer)

21CSC21 Web Programming Languages

3 0 0 3

Course Outcomes

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

1. Describe the fundamental concepts and characteristics of 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.

CO – PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	-	-	3	2
CO2	2	3	3	-	2	-	-	-	-	-	-	-	3	1
CO3	3	2	3	-	3	-	-	-	-	-	-	-	1	3
CO4	2	3	2	-	3	-	-	-	-	-	-	-	1	3
CO5	2	2	2	-	3	-	-	-	-	-	-	-	2	2
CO6	1	2	3	-	1	-	-	-	-	-	-	-	1	1

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

Syllabus

Unit I **12 Hours**

Overview of the web

Web application Overview: Basic concepts of web, The importance of web technology, 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: Overview of fundamentals of JavaScript, 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**

Web programming in Java

Introduction: Evolution of RESTful services, REST API Architectural Constraints, Designing REST API, REST API Error Handling Patterns, REST API Handling Change- Versioning Patterns, REST API Cache Control Patterns, REST API Response Data Handling Patterns, REST API Security.

Spring Boot

Introduction, Framework Theory, Beans Implementation, Dao Implementation, Running Project, Creating Model Class,

REST API Security, Creating XML Files

Unit IV **12 Hours**

Web programming in Python

Introduction: Basic concepts –, 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, ArunRavindran, 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.

21CSC22 Web Application Developments Framework

3 0 2 4

Course Outcomes

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

1. Describe the fundamentals of web frameworks and their role in web development.
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. Analyze how modern-day web applications are different from web sites.
6. Learn the technologies of Python web framework.

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	3	-	1	-	-	-	-	-	-	-	-	1
CO2	3	2	2	-	2	-	-	-	-	-	-	-	-	2
CO3	3	-	3	-	3	-	-	-	-	-	-	2	-	3
CO4	3	-	3	-	2	-	-	-	-	-	-	-	-	3
CO5	3	2	2	-	2	-	-	-	-	-	-	2	-	3
CO6	2	2	3	-	3	-	-	-	-	-	-	2	-	2

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, ShyamSeshadri, 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, DoguhanUluca, 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.

Career Path III (Cyber Security)

21ITC31 Fundamentals of Security

3 0 0 3

Course Outcomes

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

1. Explain the fundamental concepts of information security
2. Illustrate the use of cryptography and its functions
3. Identify different types of Cryptography methods
4. List out different types of Security Threats and Vulnerability
5. Make use of the concept of network security
6. Outline transport and network layer security

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	-	-	1	-	-	-	-	-	-	-	3	3
CO2	3	3	-	-	2	-	-	-	-	-	-	-	3	2
CO3	1	2	-	-	3	-	-	-	-	-	-	-	2	2
CO4	2	3	-	-	1	-	-	-	-	-	-	-	3	3
CO5	3	2	-	-	3	-	-	-	-	-	-	-	2	3
CO6	3	2	-	-	2	-	-	-	-	-	-	-	3	3

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

Unit I **12 Hours**

Security concepts

What is security, Data vs Information vs Cyber security, Goals of Information Security, Computer Security

Concepts, threats, attacks, and assets, security functional requirements, fundamental security design principles , computer security strategy, networking, benefits of networking.

protocols suite,hacking,network sniffing,social engineering

Unit II **12 Hours**

Cryptography

Introduction to cryptography, Cryptographic functions(Authentication, Nonrepudiation, confidentiality, integrity), Cryptanalysis. Types of Cryptography: Symmetric key and Asymmetric Key Cryptography, Message Integrity, Encryption and Decryption Techniques, Digital Signatures, Email Standards: MIME and PGP

Applications of Cryptography, cipher types, steganography

Unit III **12 Hours**

Security Threats and Vulnerabilities

Overview of Security Threats and Vulnerability: Types of attacks on Confidentiality, Integrity and Availability, Types of malware, Buffer Overflow

Security Counter-Measures: Passwords, Access Control (Authenticating users, Handling User Access), Antivirus Software, Firewalls, Intrusion Detection Systems and IPS.

Types of firewalls, advanced persistant threat

Unit IV **12 Hours**

Network Security

TCP/IP Suite, DNS, MITM, Attacks on TCP and DNS, VPN, Application layer security (PGP, S/MIME), Transport layer security (TLS), Network layer security (IPSec) Anonymous networks: Tor,I2P.

TCP Vulnerabilities, SSH vs SSL,

Total: 48 Hours

Textbook (s)

1. William Stallings, Lawrie Brown, Computer Security Principle sand Practice Third Edition,2015
2. Nihad A. Hassan, Rami Hijazi, Digital Privacy and Security Using Windows: A Practical Guide, Apress, 2017

Reference (s)

1. Michael T. Goodrich and Roberto Tamassia, Introduction to Computer Security, Addison Wesley, 2011.
2. Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed, 3rd Edition, John Wiley & Sons Inc, 2021.
3. Kozierok, Charles M. The TCP/IP guide: a comprehensive, illustrated Internet protocols reference, 2005
4. Harris, Shon, "CISSP all-in-one exam guide." Sixth edition (2013).

21ITC32 Cybernet Security

3 0 2 4

Course Outcomes

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

1. Explain the fundamental concepts of Cyber security
2. Demonstrate the web security and different attacks
3. Identify different network scanning and security measures
4. List out different types of Intrusion detection.
5. Model different types of Intrusion prevention systems
6. Outline different cyber-crimes, IT laws and acts.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	3	1	2	-	-	-	-	-	-	2	1	2
CO2	3	-	3	1	2	-	-	-	-	-	-	3	2	1
CO3	2	-	2	3	2	-	-	-	-	-	-	3	2	2
CO4	3	-	1	1	2	-	-	-	-	-	-	3	2	2
CO5	2	-	3	3	2	-	-	-	-	-	-	3	3	3
CO6	2	-	3	2	2	-	-	-	-	-	-	1	1	1

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

Unit I

Introduction to Cyber Security **12+8 Hours**

What is Cyber Security, its need, cyber-threats, Cyber Warfare-Cyber Crime-Cyber terrorism-Cyber Espionage), Career Growth, Statistics, Inferences, Need for a Comprehensive Cyber Security Policy, Classification of Cyber Crimes, kinds of cyber crimes, Reasons for Cyber Crimes, Cyber Security Tools : Nmap, Metasploit, wireshark, tcpdump, snort.

Cyber security awareness, social engineering, cyber stalking

Practical Components

1. Lab Activity: Study the use of network reconnaissance tools like WHOIS, dig, ping, hping, traceroute, nslookup to gather information about networks and domain registrars.
2. Lab Activity: Study of packet sniffing tools like tcpdump, Wireshark, etc.
3. Lab Activity: Perform Network Scanning using NMAP in Windows and ZENMAP in Kali Linux.
4. Lab Activity: Study of Intrusion detection tool SNORT.

Unit II

Web security **12+8 Hours**

Same origin Policy, Cross Origin Resource Sharing, DDOS, SQL Injection, XSS, Homograph, Generating and storing session tokens.

Networking Scanning & Security Measures:

Packet Sniffing and spoofing, Network scanning types, port scanning & its tools, and Network Architecture

Security Measures :IPtables(firewalls) , Webservers (Nmap & Metasploit for securing webservers), Cyber Threats and Attacks (Malware, DOS, MITM, Social engineering attacks, Spoofing, Phishing)

Cross-Site Request Forgery (XSRF/CSRF), spear phishing.

Practical Components

1. Lab Activity: Perform port scanning using Nmap.
2. Lab Activity: Penetration Testing and Exploiting with Metasploit, Armitage, and msfconsole.
3. Lab Activity: Simulate DoS and DDoS attacks using various tools.
4. Lab Activity: Study of SQLMap to explore SQL Injection attacks.

Unit III

Intrusion Detection System **12+8 Hours**

Intruders, Intrusion Detection, Analysis Approaches, Network-Based IDS, Host-Based IDS, signature based IDS, anomaly based IDS, advantages and disadvantages of NIDS and HIDS

Intrusion Detection Tools, snort architecture, snort rules, case studies of intrusion detection systems,Intrusion detection exchange format.

Honeypots, different types of honeypots, benefits and dangers of honeypots

firewall vs IDS, Physical IDS, honeynet

Practical Components

1. Lab Activity: Use iptables in Linux to create firewalls.
2. Lab Activity: Use Snort as a packet sniffer and write your own IDS rules.
3. Lab Activity: Setup honeypot.
4. Lab Activity: Demonstrate Intrusion Detection System (IDS) using any tool (SNORT or any other software).

Unit IV

Cyber Laws and Digital Forensics **12 + 8 Hours**

Digital Forensics: Introduction to Digital Forensics, historical background of digital forensics, Forensic Software, and Hardware, need for computer forensics science, special tools and techniques digital forensic life cycle, challenges in digital forensic.

Law Perspective: Introduction to the Legal Perspectives of Cybercrimes and Cybersecurity, Cybercrime and the Legal Landscape around the World, Why Do We Need Cyber laws, The Indian IT Act, Cybercrime Scenario in India, Digital Signatures and the Indian IT Act.

Cybercrime and Punishment

Practical Components

1. Lab Activity: Write a security policy for password protection.
2. Lab Activity: Install Rootkits to study a variety of options.
3. Lab Activity: Implement a Digital Signature Scheme.
4. Lab Activity: Case Study on Indian IT Act 2000.

Total: 48+32 Hours

Textbook (s)

1. Wenliang Du, Computer & Internet Security: A Hands-on Approach, (2020)
2. William Stallings, Lawrie Brown, Computer Security Principle sand Practice Third Edition,2015
3. SunitBelapure and Nina Godbole, Cyber Security: Understanding Cyber Crimes, Computer Forensics And Legal Perspectives, Wiley India Pvt. Ltd, 2011.
4. Nelson Phillips and EnfingerSteuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi

Reference (s)

1. Pande, Jeetendra. "Introduction to Cyber Security.", (2017)
2. Pavan Duggal, Cyber frauds, cybercrimes & law in India.
3. Ali A. Ghorbani, Network intrusion detection and prevention concepts and techniques, Springer, 2010
4. Roberto Di Pietro, Luigi V. Mancini (2008), Intrusion Detection System, Springer
5. DafyddStuttard and Marcus Pinto. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Wiley Publication

7th Semester

Career Path I (Artificial Intelligence& Machine Learning)

21CSC13 Natural Language Processing

3 0 0 3

Course Outcomes

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

1. Describe the fundamentals and building blocks of Natural Language Processing (NLP).
2. Explain different text representation techniques and labeling methods in NLP.
3. Apply different Natural Language models and named entities
4. Explain the fundamentals of RNN and their applications in NLP.
5. Make use of GRUs and LSTM models for translation
6. Describe various applications of NLP across different domains.

CO–PO Mapping:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	-	-	-	-	-	-	-	2	-	2
CO2	3	2	2	2	-	-	-	-	-	-	-	2	-	3
CO3	3	3	2	1	-	-	-	-	-	-	-	3	-	3
CO4	3	3	2	3	-	-	-	-	-	-	-	3	-	3
CO5	3	3	2	3	-	-	-	-	-	-	-	3	-	3
CO6	3	3	2	3	-	-	-	-	-	-	-	3	-	3

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

Unit I **12 Hours**

Introduction to NLP: Definition, History, NLP in the real world, Building blocks of language, Approaches to NLP, NLP Pipeline, NLP Challenges, Deep learning in NLP, Morphology fundamentals;

Objectives of NLP, The applications of NLP.

Unit II **12 Hours**

Text Representation and Sequence Labelling

Basic Vectorization approaches- One-Hot Encoding, Bag of Words, Bag of N-Gram, TF-IDF; Distributed universal text and handcrafted feature Representations, Neural language models, N-gram language model Sequence labeling for POS and Named Entities: POS tagging, Named Entities tagging, hidden markov models, conditional Random Fields Visualizing Embeddings, Viterbi algorithm.

Evaluation of Named Entity Recognition, Markov chains.

Unit III **12 Hours**

Deep learning architectures for NLP:

RNN for language model, Sequence Labeling and Sequence Classification, Encoder-Decoder with RNNs and Transformers, GRUs and LSTMs for machine translation, Convolutional neural networks for sentence classification.

Transformers as Autoregressive Language Models, Potential Harms from Language Models.

Unit IV **12 Hours**

Case Study on NLP:

sentiment analysis, machine translation, automated speech recognition systems, question-answering based systems, topic modelling, Text Generation and Summarization.

Semantic Role LabelingLexicons for Sentiment, Affect, and Connotat.

Total: 48 Hours

Textbook (s)

1. Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft)
2. Yoav Goldberg. Neural Network Methods for Natural Language Processing
3. Vajjala, Sowmya, BodhisattwaMajumder, Anuj Gupta, and HarshitSurana. Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. O'Reilly Media, 2020.

Reference (s)

1. Rajesh Arumugam, RajalingappaaShanmugamani :Hands-On Natural Language Processing with Python
2. Manning C.,SchützeH ,Foundations of Statistical Natural Language Processing–.(MITPress)
3. Jacob Eisenstein. Natural Language Processing.

Career Path II (Full Stack Developer)

21CSC23 Web Application Databases

3 0 0 3

Course Outcomes

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

1. Describe the working principles of web-based client-server applications.
2. Analyze architecture of various web-based applications.
3. Examine the requirements of structured and un-structured data.
4. Identify the structure of modern-day web applications.
5. Identify the use advanced querying techniques to enable faster data transfer.
6. Analyze the security issues in data transfer via the web.

CO – PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	-	1	-	-	-	-	-	-	2	-	1
CO2	3	2	2	-	1	-	-	-	-	-	-	2	-	3
CO3	2	3	3	-	3	-	-	-	-	-	-	1	-	3
CO4	2	2	2	-	3	-	-	-	-	-	-	2	-	3
CO5	1	2	2	-	3	-	-	-	-	-	-	2	-	3
CO6	2	2	2	-	3	-	-	-	-	-	-	2	-	3

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

Syllabus

Unit I **12 Hours**

Overview of Web databases

Web application Overview: Basic concepts of web, Web database architectures, Web database gateways, Web-database programming: Client-side and server-side web database programming

Database gateways, CGI, Browser extensions

Unit II **12 Hours**

Structured and unstructured Web databases

Structured databases: Structured data, Data models, Relational database management systems, MySQL – Introduction, querying, updating and deleting data, Challenges of structured data.

Unstructured databases: Unstructured data and its usages, NoSQL databases, MongoDB- Introduction, querying, updating and deleting data, Challenges of unstructured data- Cassandra, GraphQL

Structured data, unstructured data, Relational database, Document-based database

Unit III **12 Hours**

Database connections & data operations

Connecting to database: Native database APIs, Database-independent APIs, Template-driven database access packages, Third-party class libraries.

CRUD operations: Querying web database, User driven querying, writing to web databases – create, delete and update.

JDBC, ODBC, template parsing.

Unit IV **12 Hours**

Authentication, Authorization and Communication in web services

Web requests: Structure of web requests, JavaScript Object Notation (JSON), Request methods – GET, POST, PUT, DELETE, response status codes

State management: Session management, cookies, request and response headers.

Security: User authentication, User authorisation, Proxy servers, Digital signatures, Digital certificates, SSL and HTTPS.

Kafka, RabbitMQ

JSON, User authentication and authorisation

Total: 48 Hours

Textbook (s)

1. Web Database Applications with PHP and MySQL, Hugh E. Williams, David Lane, 2nd Edition, O'Reilly, 2004.
2. NoSQL Distilled, PramodSadalage, Martin Fowler, 1st Edition, 2012

Reference (s)

1. Getting Started with NoSQL, GauravVaish, Paperback, Packt, 2013.
2. Fundamentals of Database Systems, ElmasriNavathe Pearson Education.
3. An Introduction to Database systems, C.J. Date, A.Kannan, S.SwamiNadhan, Pearson, 8th Edition.
4. Web Application Security, Andrew Hoffman, O'Reilly, 2020.

Career Path III (Cyber Security)

21ITC33 Cloud Security

3 0 0 3

Course Outcomes

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

1. Understand core concepts of the cloud computing paradigm
2. Learn about characteristics, advantages and challenges brought by the various models and services in cloud computing.
3. Identify the known threats, risks, vulnerabilities and privacy issues associated with Cloud based services.
4. Understand security challenges, threats and risks involved in the cloud application
5. Anlayze the Security Requirements for the Architecture in cloud services
6. Understand Data Security and best practices involved in securing the cloud.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	-	1	2	-	-	-	-	-	-	-	-	-
CO2	2	2	-	2	3	-	-	2	-	-	-	-	-	-
CO3	2	1	-	2	2	-	-	2	-	-	-	-	-	-
CO4	3	2	-	2	3	-	-	2	-	-	-	-	-	-
CO5	2	3	-	3	2	-	-	-2	-	-	-	-	-	-
CO6	3	2	-	2	2	-	-	2	-	-	-	-	-	-

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

Unit I **11 Hours**

Cloud Computing Fundamentals and Architecture- Understanding and Roots of Cloud Computing, Essential Characteristics, Cloud Reference Architecture, Cloud Service Models: SaaS, Paas, IaaS, Cloud Deployment Models, Expected Benefits, Forming Clouds with example.

Unit II **13 Hours**

Cloud Computing Software Security Fundamentals- Cloud Information Security Objectives, Cloud Security Services, Relevant Cloud Security Design Principles,

Secure Cloud Software Requirements, Security Concerns, Risk Tolerance, Legal and Regulatory Issues

Unit III **12 Hours**

Cloud Computing Risk Issues and Security Challenges:

The CIA Triad, Privacy and Compliance Risks, Threats to Infrastructure, Data, and Access Control, Cloud Service Provider Risks

Security Challenges- Security Policy Implementation, Virtualization Security Management, VM Security Recommendations, VM-Specific Security Techniques.

Unit IV **12 Hours**

Securing the cloud: Architecture and Data Security

Security Requirements for the Architecture,Security Patterns and Architectural Elements,Cloud Security Architecture,Planning Key Strategies for Secure Operation .

Overview of Data Security in Cloud Computing,Data Encryption: Applications and Limits,Cloud Data Security: Sensitive Data Categorization,Cloud Data Storage,Cloud Lock-in, Key strategies to secure the cloud,Best practices for cloud computing,security monitoring

Total: 48 Hours

Textbook (s)

1. Russell Dean Vines and Ronald L. Krutz, Cloud Security: A Comprehensive Guide To Secure Cloud Computing, Wiley India Pvt Ltd, 2010
2. Vic Winkler (J.R.), Securing The Cloud: Cloud Computing Security Techniques and Tactics, Syngress/Elsevier ,2011

Reference (s)

1. Thomas Erl, Cloud Computing Design Patterns, Prentice Hall, 2015
2. Barrie Sosinsky, Cloud Computing Bible, Wiley India, 2011
3. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2011

21IT010 Social Network Analysis (Elective V)

3 0 0 3

Course Outcomes

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

1. Acquire Knowledge to analyze Social Networks
2. Model, Aggregate and Represent Knowledge for Semantic Web
3. Use Extraction and Mining tools for Social Networks
4. Apply Reality Mining to Predict Human Behaviors for Social Communities
5. Apply various Algorithms for Evolution and Opinion Mining in Social Networks
6. Write algorithms and systems for expert location in social networks

CO – PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	2	3	-	-	-	-	-	-	-	-	-
CO2	3	2	2	2	3	-	-	-	-	-	-	-	-	-
CO3	3	2	2	2	3	-	-	-	-	-	-	-	-	-
CO4	3	2	2	2	2	-	-	-	-	-	-	-	-	-
CO5	3	2	2	2	3	-	-	-	-	-	-	-	-	-
CO6	3	2	2	2	2	-	-	-	-	-	-	-	-	-

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

Syllabus

Unit I **11 Hours**

Introduction

Introduction to Web, Limitations of current Web, Development of Semantic Web, Emergence of the Social Web, Statistical Properties of Social Networks, Network analysis, Development of Social Network Analysis-Key concepts and measures in network analysis, Discussion networks-Blogs and online communities-Web-based networks

Case Studies of Social Network sites like Facebook-Twitter-Linkedin etc.

Unit II **13 Hours**

Evolution

Evolution in Social Networks– Framework, Tracing Smoothly Evolving Communities, Models and Algorithms for Social Influence Analysis, Influence Related Statistics, Social Similarityand Influence, Influence Maximization in Viral Marketing, Link Prediction in Social Networks, Feature based Link Prediction

Neurons &Social Actor-Advanced techniques in Link Prediction

Unit III **12 Hours**

Mining Communities and Opinion Mining

Applications of Community Mining Algorithms, Node Classification in Social Networks

Opinion Extraction–Sentiment Classification and Clustering, Temporal Sentiment Analysis-Irony Detection in Opinion Mining-Wish Analysis–Product Review Mining–Review Classification

Tracking Sentiments towards Topics over Time

Unit IV **12 Hours**

Modelling and Visualization

Visualizing Online Social Networks, A Taxonomy of Visualizations, Graph Representation-Centrality-Clustering-Node-Edge Diagrams-Visualizing Social Networks with Matrix-Based Representations-Node-

Link Diagrams, Hybrid Representations, Modelling and Aggregating Social Network Data, Random Walks and their Applications, Ontological representation of Social Individuals and Relationships.
Use of Hadoop and MapReduce-Gephi

Total: 48 Hours

Textbook (s)

1. Charu C. Aggarwal, Social Network Data Analytics, Springer, 2011

2. Peter Mika, Social Networks and the Semantic Web, Springer, 1st Edition, 2007.

Reference (s)

1. BorkoFurht, Handbook of Social Network Technologies and Applications, Springer, 1st Edition, 2010.
2. GuandongXu,Yanchun Zhang and Lin Li, Web Mining and Social Networking, Techniques and applications, Springer, 1st Edition, 2011.
3. Giles, Mark Smith, John Yen, Advances in Social Network Mining and Analysis, Springer,2010.
4. Ajith Abraham, Aboul Ella Hassanien, VáclavSnášel, Computational Social NetworkAnalysis: Trends, Tools and Research Advances, Springer, 2009.

21CS011 Optimization Techniques (Elective V)

3 0 0 3

Course Outcomes

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

1. Explain the need of optimization of engineering systems.
2. Formulate and solve linear programming problems.
3. Solve the problems on network models such as transportation and assignment.
4. Analyze the complexity of solutions to an optimization problem.
5. Apply unconstrained optimization and constrained non-linear programming
6. Illustrate about dynamic programming and modern optimization techniques.

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	-	2	-	-	-	-	-	-	2	-	-
CO2	2	2	1	-	1	-	-	-	-	-	-	2	-	-
CO3	2	3	2	-	2	-	-	-	-	-	-	2	-	-
CO4	2	2	1	-	1	-	-	-	-	-	-	2	-	-
CO5	2	1	3	-	3	-	-	-	-	-	-	2	-	-
CO6	2	3	2	-	2	-	-	-	-	-	-	2	-	-

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

Unit I **12 Hours**

Introduction to Optimization & Classical Optimization Techniques:

Engineering Applications of Optimization, Statement of Optimization Problem.

Classical Optimization Techniques: Single variable optimization, multivariable optimization with no constraints, multivariable optimization with equality constraints, multivariable optimization with inequality constraints, convex programming problem.

Optimization techniques, Engineering Optimization Literature.

Unit II **12 Hours**

Linear Programming & Transportation Problems:

Applications of Linear Programming, Standard form of a linear programming problem – geometry of linear programming problems – definitions and theorems – solution of a system of linear simultaneous equations – pivotal reduction of a general system of equations – simplex algorithm.

Introduction to Transportation problems, various methods of Transportation problem, Variations in Transportation problem,

Two phases of simplex method

Unit III **12 Hours**

Assignment Problems & Unconstrained Nonlinear Programming:

Introduction to Assignment problems, variations in Assignment problems. Transportation Problem: Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel's approximation method – testing for optimality of balanced transportation problems.

Introduction, Random Search Methods, Grid Search method, Univariate method, Powell's Method, Simplex Method, Gradient of a Function, Steepest descent method. Newton's Method.

Marquardt Method, Quasi-Newton Methods

Unit IV **12 Hours**

Dynamic Programming & Modern methods of Optimization:

Multistage decision process, Concept of suboptimization and principle of optimality, computational procedure in dynamic programming, Examples of calculus and tabular method of solutions.

Modern methods of optimization: Genetic Algorithms, particle swarm optimization and Ant colony optimization.

Simulated Annealing, neural network-based optimization

Total: 48 Hours

Textbook (s)

1. Singiresu S. Rao, Engineering Optimization: Theory and Practice, Wiley, 4th edition, 2009.

Reference (s)

1. Deb K. - Optimization for Engineering Design Algorithms and Examples, PHI - 2000
2. George Bernard Dantzig, MukundNarainThapa, Linear programming, Springer series in operations research 3rd edition, 2003.
3. H.A. Taha, Operations Research: An Introduction, 8th Edition, Pearson/Prentice Hall, 2007.
4. Kalyanmoy Deb, Optimization for Engineering Design-Algorithms and Examples, PHI Learning Pvt. Ltd., New Delhi, 2005.
5. Arora J, Introduction to Optimization Design, Elsevier Academic Press, New Delhi, 2004
6. Saravanan R, Manufacturing Optimization through Intelligent Techniques, Taylor & Francis (CRC Press), 2006
7. Hardley G., Linear Programming, Narosa Book Distributors Private Ltd., 2002

Total: 48 Hours

Textbook(s)

1. AdHoc Wireless Networks: Architectures and Protocols-C.Siva Ram Murthy and B.S.Manoj,2004,PHI.
2. WirelessAd-hocandSensorNetworks:Protocols,PerformanceandControl-JagannathanSarangapani,CRCPress

Reference(s)

1. WirelessSensorNetworks-C.S.Raghavendra,KrishnaM.Sivalingam,2004,Springer.
2. Ad-HocMobileWirelessNetworks:Protocols&Systems,C.K.Toh,Firsted.PearsonEducation

3 0 0 3

21CS014 Green Computing (Elective VI)

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

1. Describe green IT with its different dimensions and strategies.
2. Analyze green devices and hardware along with its green software methodologies.
3. Examine the green IT standards for enhancing environmental sustainability.
4. Analyze the various green enterprise activities, functions and their role with IT.
5. Identify the various laws, standards and protocols for regulating green IT.
6. Identify the various key sustainability and green IT trends.

CO – PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	-	3	3	-	-	-	-	-	-	-
CO2	3	3	3	-	-	3	3	-	-	-	-	-	-	-
CO3	3	3	3	-	-	3	3	-	-	-	-	-	-	-
CO4	3	3	3	-	-	3	3	-	-	-	-	-	-	-
CO5	3	2	2	-	-	3	3	-	-	-	-	-	-	-
CO6	3	2	2	-	-	3	2	-	-	-	-	-	-	-

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

Syllabus

Unit I **12 Hours**

Fundamentals

Introduction, Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Green IT, Holistic Approach to Greening IT, Greening IT, Applying IT for enhancing Environmental Sustainability, Green IT Standards and Eco-Labeling of IT.

Enterprise Green IT strategy, Green IT: Burden or Opportunity?

Unit II **12 Hours**

Green Devices and Hardware with Green Software

Green Devices and Hardware: Introduction, Life Cycle of a device or hardware, Reuse, Recycle and Dispose. Green Software: Introduction, Energy-saving software techniques, Evaluating and measuring software Impact to platform power.

Research the sustainability of paper use in an organization.

Unit III **12 Hours**

Green Enterprises and the Role of IT

Introduction, Organization and Enterprise Greening, Information systems in Greening Enterprises, Greening Enterprise: IT Usage and Hardware, Inter-Organizational Enterprise activities and Green Issues, Enablers and making the case for IT and Green Enterprise.

Analyze an organization's IT sustainability.

Unit IV **12 Hours**

Managing and Regulating Green IT

Introduction, Strategizing Green Initiatives, Implementation of Green IT, Information Assurance, Communication and Social media.

Laws, Standards and Protocols : Introduction, The regulatory environment and IT manufacturers, Non regulatory government initiatives, Industry associations and standards bodies.

Green building standards, Green data centers, Social movements and Greenpeace.

Total: 48 Hours

Textbook (s)

1. Harnessing Green IT Principles and Practices, San Murugesan, G.R. Gangadharan, Wiley Publication, ISBN:9788126539680,2012.

Reference (s)

1. Foundation of Green IT, Marty Poniatowski, Prentice Hall Publication, ISBN:9780137043750,2009.

21CS015 Software Project Management (Elective VI)

3 0 0 3

Course Outcomes

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

1. Explain the concepts and issues of Software Project Management
2. Illustrate various Software Architectures
3. Classify Software Risks and Risk Management Strategies
4. Design effective software development model to meet organizational needs
5. Experiment with appropriate methodologies to develop a project schedule
6. Experiment with appropriate techniques to assess ongoing project performance

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	2	2	-	2
CO2	3	2	2	-	-	-	-	-	-	-	2	2	-	3
CO3	3	2	2	-	-	-	-	-	-	-	2	2	-	3
CO4	3	2	3	-	-	-	-	-	-	-	2	2	-	3
CO5	3	2	3	-	-	-	-	-	-	-	3	2	-	3
CO6	3	2	3	-	-	-	-	-	-	-	3	2	-	3

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

Unit I **12 Hours**

Conventional Software Management: The Waterfall Model, Conventional Software Management Performance. **Transition:** The principles of conventional software Engineering, Principles of Modern software management, transitioning to an iterative process.

Evolution of Software Economics: Software Economics, Software Cost Estimation. **Improving Software Economics:** Reducing Software Product Size, Improving Software Processes, Improving Team Effectiveness, Improving Automation, Achieving Required Quality, Peer Inspections.

Software Development Phases, Procedural vsObject Oriented Paradigm.

Unit II **12 Hours**

Life Cycle Phases: Engineering and Production Stages, Inception, Elaboration, Construction, Transition Phases.

Artifacts of Software Process: The Artifact Sets, Types of Artifacts.

Software Architectures: A Management Perspective and Technical Perspective,

Workflows of the process: Software Process Workflows, Iteration Workflows.

Off-the-Shelf Components, Requirement Specification Document, Methods of Specification.

Unit III **12 Hours**

Checkpoints of the Process: Major Milestones, Minor Milestones, Periodic Status Assessments.

Iterative Process Planning: Work breakdown Structures, Planning Guidelines, Cost and Schedule Estimating process, Iteration Planning Process.

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, Evolution of Organizations.

Process Automation: Automation Building Blocks, The Project Environment.

CASE Tools for Requirements Specification, Planning and Estimation, Pragmatic Planning.

Unit IV **12 Hours**

Project control and Process Instrumentation: The Seven Core Metrics, Management Indicators, Quality Indicators, Life Cycle Expectations, Pragmatic Software Metrics, Metrics Automation, Tailoring the Process.

Future Software Project Management: Modern Project Profiles, Next Generation Software Economics,The COCOMO Cost Estimation Model: Basic COCOMO

Bugs of testing, Bug tracking tools

Total: 48 Hours

Textbook (s)

1. Walker Royce, “Software Project Management – A UnifiedFramework”, 1stEdition, Pearson Education, 2005.
2. PankajJalote, “Software Project Management in Practice”, 1stEdition, Pearson Education, 2002.

Reference (s)

1. Bob Hughes, “Mike Cotterell, Rajib Mall, Software ProjectManagement”, 5thEdition, McGraw-Hill Higher Education, 2011.
2. Joel Henry, “Software Project Management”, 1st Edition, Pearson Education, 2004.
3. Norman E. Fenton,Shari Lawrence Pfleeger, “Software Metrics: A Rigorous and Practical Approach “, 1st Edition, PWS Publishing Company, 1997.

21CS016 Soft Computing (Elective VI)

3 0 0 3

Course Outcomes

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

1. Identify an appropriate soft computing technique to build an Intelligent Machine
2. Apply supervised learning methods for neural network models for pattern classification and regression problems
3. Demonstrate the use of various Associative Memory Networks
4. Apply unsupervised learning methods of neural network models for storing and pattern classification
5. Determine the solutions of modeled networks using soft computing approaches for a given problem
6. Explain fuzzy logic and Genetic Algorithm to handle uncertainty

COs – POs Mapping

CO’s	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	2	-	-	-	-	-	-	-	2	2	-
CO2	2	3	2	2	-	-	-	-	-	-	-	2	2	-
CO3	2	2	3	3	-	-	-	-	-	-	-	2	2	-
CO4	3	3	2	2	-	-	-	-	-	-	-	2	2	-
CO5	3	3	2	2	-	-	-	-	-	-	-	2	2	-
CO6	3	2	3	3	-	-	-	-	-	-	-	2	2	-

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

Unit I **12 Hours**

Basic elements of Soft Computing: Difference between Hard Computing & Soft Computing - Introduction to Neural Networks - Fuzzy Logic - Genetic Algorithms - Hybrid Systems.

Artificial Neural Networks: Introduction to Artificial Neural Networks - Evolution of Neural Networks - Basic Models of Neural Networks - McCulloch-Pitts Neuron - Hebb Network.

Basics of Artificial Intelligence and Computational Intelligence.

Unit II **12 Hours**

Supervised Learning Network: Artificial Neural Networks: Introduction - Perceptron Networks - Back Propagation Network. Radial Basis Function Network.

Associative Memory Networks: Auto-associative Memory Network - Hetero auto-associative Memory Network - Bidirectional Associative Memory (BAM) - Hopfield Networks.

Functional Link Neural Network.

Unit III **12 Hours**

Unsupervised Learning Network: Introduction - Fixed Weight Competitive Nets -Kohonen Self-Organizing Feature Maps - Learning Vector Quantization - Counter Propagation Networks – Basic Concepts on Adaptive Resonance Theory Networks (ART-1 & ART-2).

Support Vector Machines and their applications for Classification.

Unit IV **12 Hours**

Introduction to Fuzzy Sets: Difference between Classical Sets & Fuzzy Sets – Properties.

Classical Relations and Fuzzy Relations: Cartesian product of Relations - Classical Relations - Fuzzy Relations

Membership Functions: Features of Membership Functions -Fuzzification - and Defuzzification

Genetic Algorithm: Basic Concepts on Genetic Operators.

Concepts on Fuzzy Controller and its applications - a Simple Application of Genetic Algorithm for function Optimization.

Total: 48 Hours

Textbook (s)

1. S. N. Sivanandam, S N Deepa, Principles of Soft Computing, 2nd Edition, Wiley India, 2007
2. V. Kecman, Learning and Soft Computing, Pearson Education, India, 2006

Reference(s)

1. J-S R Jung, C-T Sun and E. Mizutani, Neuro Fuzzy and Soft Computing, Pearson Education, India, 2015
2. G. J. Klir and Bo Yuan, Fuzzy sets and Fuzzy Logic, Prentice Hall, USA, 1995
3. D. E. Goldberg, Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley, N.Y, 1989
4. S. Haykins, Neural networks: a comprehensive foundation Pearson Education, India, 2002
5. A. P. Engelbrecht, Computational Intelligence: An Introduction, John Wiley & Sons, 2007.
6. X. Yu and M. Gen, Introduction to Evolutionary Algorithms, Springer Verlag, 2010.

21SIX02 Summer Internship II

0 0 0 1.5

Course Outcomes

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

1. Demonstrate communication skills to meet the requirement of industry
2. Develop logical thinking and analytical skills to thrive in competitive examinations
3. Use mathematical concepts to solve technical quizzes
4. Develop technical skills to work out real time problems
5. Develop algorithms for different applications
6. Solve industry defined problems using appropriate programming skills

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	2	1	2	3	1	3	3	3	1	3	3	1
CO2	3	3	3	3	2	2	1	3	3	1	1	3	3	2
CO3	3	2	3	2	2	1	1	3	2	3	1	2	3	1
CO4	3	3	3	3	3	3	2	3	3	3	1	3	3	2
CO5	3	3	3	3	3	3	3	3	3	3	2	3	3	2
CO6	3	3	3	3	3	3	3	2	3	3	3	3	3	2

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

21PWX01Project Work

0 0 16 8

Course Outcomes

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

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.

CO–PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
1	3	3	2	3	2	3	3	2	3	3	3	3	3	3
2	3	3	3	3	3	3	2	3	3	3	3	3	3	3
3	3	3	3	3	2	3	2	3	2	3	3	2	3	3
4	3	3	3	3	2	2	3	3	2	3	3	2	3	3
5	3	3	3	3	3	3	3	3	3	3	3	3	3	3
6	3	3	3	3	3	3	3	3	3	3	3	3	3	3

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

8th Semester

21CS017 Fundamentals of Social Networks Analysis (Elective VIII)

7.
0 0 0 3

Course Outcomes

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

1. Explain the fundamental principles of social networking
2. Identify the architectures and challenges in building social networks
3. Interpret the capabilities and limitations of the existing network analysis methods
4. Model the knowledge to analyze real-world networks
5. Demonstrates the knowledge of basic mathematical models used in the analysis of social networks
6. Demonstrate human behaviour in social web and related communities.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	1	1	2	-	-	-	-	-	-	-	-	2
CO2	3	3	1	1	1	-	-	-	-	-	-	-	-	2
CO3	3	1	2	3	3	-	-	-	-	-	-	-	-	2
CO4	3	3	2	1	1	-	-	-	-	-	-	-	-	2
CO5	3	1	2	3	3	-	-	-	-	-	-	-	-	2
CO6	3	1	2	2	2	-	-	-	-	-	-	-	-	2

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

Unit I **11 Hours**

Introduction to social network analysis and Descriptive network analysis: Introduction to new science of networks. Networks examples. Graph theory basics. Statistical network properties.Degree distribution, clustering coefficient. Frequent patterns.Network motifs.Cliques and k-cores.

Unit II **13 Hours**

Network structure, Node centralities and ranking on network: Nodes and edges, network diameter and average path length.
Node centrality metrics: degree, closeness and betweenness centrality. Eigenvector centrality and PageRank.Algorithm HITS.

Unit III **12 Hours**

Network communities and Affiliation networks: Networks communities.
Graph partitioning and cut metrics. Edge betweenness.Modularity clustering.
Affiliation network and bipartite graphs.1-mode projections. Recommendation systems

Unit IV **12 Hours**

Predicting human behaviour and privacy issues: Understanding and predicting human behaviour for social communities - User data management - Inference and Distribution - Enabling new human experiences - Reality mining - Context - Awareness - Privacy in online social networks - Trust in online environment - Trust models based on subjective logic - Trust network analysis - Trust transitivity analysis - Combining trust and reputation - Trust derivation based on trust comparisons - Attack spectrum and countermeasures.

Applications of SNA: covert networks , community welfare, collaborative networks, co-citation networks

Total: 48 Hours

Textbook (s)

1. David Easley and John Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press 2010.
2. Stanley Wasserman and Katherine Faust, Social Network Analysis. Methods and Applications, Cambridge University Press, 1994

Reference (s)

1. Matthew O. Jackson, Social and Economic Networks, Princeton University Press, 2010.
2. Eric Kolaczyk, Gabor Csardi, Statistical Analysis of Network Data with R, Springer, 2014.
3. Mark Newman, Networks: An Introduction, Oxford University Press, 2010.

21CS018 Information Retrieval Systems (Elective VIII)

0 0 0 3

Course Outcomes

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

1. Explain the Objectives of Information Retrieval Systems and its relation to Digital Libraries and Data warehouse
2. Illustrate various System Search Browse and Miscellaneous Capabilities
3. Explain Automatic Indexing, Information Extraction methods
4. Demonstrate Inverted file structures, N-gram data structure, and PAT data structure
5. Interpret Classical techniques of Information Retrieval, and additional techniques employed by Web search engines
6. develop information retrieval system by using user search techniques and text search Algorithms

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	2	2	-	-	-	-	-	-	2	-	-
CO2	3	2	2	2	2	-	-	-	-	-	-	2	-	-
CO3	2	3	1	1	3	-	-	-	-	-	-	2	-	-
CO4	2	3	1	1	2	-	-	-	-	-	-	2	-	-
CO5	2	3	1	1	3	-	-	-	-	-	-	2	-	-
CO6	3	3	2	2	3	-	-	-	-	-	-	2	-	-

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

Unit I **11 Hours**

Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital libraries and Data Warehouses.

Information Retrieval System Capabilities: Search, Browse, Miscellaneous.

Miscellaneous capabilities.

Unit II **13 Hours**

Cataloging and Indexing: Objectives, Indexing Process, Automatic Indexing, Information Extraction.

Data Structures: Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure.

Signature file structure, Hypertext data structure.

Unit III **12 Hours**

Automatic Indexing: Classes of automatic indexing, Statistical indexing, Natural language, Conceptindexing, Hypertext linkages

Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters.

Measurement Example-TREC Results.

Unit IV **12 Hours**

User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, Searching the Internet and hypertext, Information Visualization

Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems.

Hardware text search systems

Total: 48 Hours

Textbook (s)

1. Kowalski, Gerald, Mark T Maybury: Information Retrieval Systems: Theory and Implementation, Kluwer Academic Press, 1997.
2. http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-in-information-retrieval-1.html

Reference (s)

1. Frakes, W.B. and RicardoBaeza Yates, Information Retrieval Data Structures and Algorithms, 1st Edition Prentice Hall, 1992.
2. Robert Korfhage, Information Storage & Retrieval, 1st Edition, John Wiley & Sons, 2005

21CS019 Fundamentals of DevOps(Elective VIII)

0 0 0 3

Course Outcomes

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

1. Illustrate the need of Improvement and value of DevOps
2. Outline the Value Stream using DevOps
3. Describe the Organizational Change and transformation
4. Illustrate the Concept and Goal of Accelerate Flow
5. Outline Feedback Loops and its usage
6. Demonstrate the Concept and Goal of Learning

CO-PO Mapping:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	-	-	-	-	-	-	-	3	-	-	3
CO2	3	3	2	-	-	-	-	-	-	-	2	-	-	3
CO3	3	2	2	-	-	-	-	-	-	-	2	-	-	3
CO4	3	2	3	-	-	-	-	-	-	-	2	-	-	3
CO5	3	3	3	-	-	-	-	-	-	-	2	-	-	3
CO6	3	2	3	-	-	-	-	-	-	-	2	-	-	3

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

UNIT I **12 Hours**

Introduction to DevOps: Understanding Improvement, The Convergence, History, and Value of DevOps; **Understanding the Value Stream using DevOps:**Analyzing the technology Value Stream; The ways of DevOps: Flow, Feedback Loops, Culture of Continual Experimentation and Learning;

UNIT II **12 Hours**

Value Stream in DevOps: Concept and Goals of Streams, Picking a Value Stream, Understanding Organizational Change, Enabling Transformation;

UNIT III **13 Hours**

Accelerate Flow: Concept and Goal of Accelerate Flow, Continuous Delivery Patterns and Practices, the Deployment Pipeline, Creating Consistency in the Pipeline, Automated Testing, Deployment Strategies; **Amplifying the Feedback Loops:** Concept and Goals of Feedback Loops,Creating a Service Reliability Culture, Fast Feedback, Understanding Monitoring, Understanding Complexity

UNIT IV **11 Hours**

Learning Acceleration with DevOps: Concept and Goal of Learning, Learning Organizations, Communication, Blameless Culture;

Total: 48 Hours

Text Books:

1. Kim, G., Behr, K., and Spafford, G. (2013). The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. IT Revolution Press
2. Kim, G., Humble, J., Debois, P., and Willis, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press

Reference Books:

1. JulianFish ,ThePracticalGuide toEnterpriseDevOps andContinuousDelivery

21FIX01 Full Semester Internship (FSI)

0 0 0 9

Course Outcomes

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

1) Use the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
2) Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3) Select appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
4) Use ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
5) Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
6) 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

COs-POs Mapping:

Cos	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	3	2	3	2	2	2	2	2	2	2	3	3
CO2	3	3	3	3	3	2	2	2	2	2	2	2	3	3
CO3	3	3	3	2	3	3	3	2	2	2	3	2	3	3
CO4	2	2	2	3	3	3	3	2	2	2	3	2	2	3
CO5	2	2	3	3	3	3	3	2	2	3	3	3	3	3
CO6	3	3	3	3	3	3	3	2	2	3	3	3	3	3

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

B. Tech. (Honors)

Domain I (Data Engineering)

21CSH11 Advanced Data Structures

4 0 0 4

Course Outcomes

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

1. Recognize the variations in implementation of fundamental linear data structures
2. Describe and implement different types of search and balanced trees
3. Demonstrate the concepts and implementations of heaps and its types
4. Identify heap data structure for appropriate use cases
5. Outline the applications of hash table data structure
6. Illustrate the use of data structure for strings

CO-PO Mapping

CO’s	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	-	-	-	-	-	-	-	-	2		2
CO2	3	2	1	-	-	-	-	-	-	-	-	2	-	2
CO3	3	2	1	-	-	-	-	-	-	-	-	2	-	2
CO4	3	2	2	-	-	-	-	-	-	-	-	1	-	2
CO5	3	2	2	-	-	-	-	-	-	-	-	2	-	2
CO6	3	2	2	-	-	-	-	-	-	-	-	2	-	2

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

Unit I

Elementary Data Structures: Review **11Hours**

List – Stack – Queue – Double Ended Queue – Dynamic Allocation of nodes – Shadow copies of Array based structures – Generic List

Unit II

Search Trees and Balanced Trees **13Hours**

Two models of search trees – General properties and Transformations – Height of search tree – Basic insert, delete operations – Dealing with non-unique keys – Building Optimal search trees – Converting trees into lists – Height balanced trees – Weight balanced trees – Red-Black trees – Splay trees – Skip lists – Interval trees – Segment trees

Unit III

Heaps **12 Hours**

Balanced search trees as heaps – Array based heaps – Heaps as trees – Leftlist heaps – Skew heaps – Binomial heaps – Changing keys in heaps – Fibonacci heaps – Heaps of optimal complexity – Double ended heap structures

Unit IV **12 Hours**

Hash Tables and Data Structures for Strings

Hash Tables

Basic hash tables and collision resolution – Universal families of hash functions – perfect hash functions – hash trees – Extendible hashing – Bloom filters

Tries and Compressed tries – Dictionaries – Suffix trees – Suffix arrays

Total: 48 Hours

Textbook (s)

1. Peter Brass, “Advanced Data Structures”, Cambridge University Press, 2008
2. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, “Introduction to Algorithms”, MIT Press, Cambridge, 2009

Reference (s)

1. Alfred L Aho, John E Hopcroft, Jeffery D Ullman, "Data Structures and Algorithms, Pearson Publications, 2016
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Pearson Education, 2002

3. Michael Main, Walter Savitch, Data Structures and other objects using C++, 4th Edition, Addison Wesley, 2018

21CSH14 Bioinformatics

4 0 0 4

Course Outcomes

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

1. Explain different types of bioinformatics data (gene, protein, disease, etc.), including their biological characteristics and relationships
2. Discuss the differences between genomics and proteomics
3. Analyse processed data with the support of analytical and visualization tools
4. Apply advisement, including systems biology, structural bioinformatics and proteomics
5. Analyse how to solve the biological problems using computational approach
6. Analyse internet packages of bioinformatics.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	1	2	-	-	-	-	-	-	-	-	2
CO2	3	2	2	1	2	-	-	-	-	-	-	-	-	2
CO3	3	2	1	1	3	-	-	-	-	-	-	-	-	2
CO4	3	2	3	1	3	-	-	-	-	-	-	-	-	3
CO5	3	3	3	1	3	-	-	-	-	-	-	-	-	3
CO6	2	3	2	1	2	-	-	-	-	-	-	-	-	3

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

UNIT –I **12 Hours**

Introduction: Definitions, Sequencing, Biological sequence/structure, Genome Projects, Pattern recognition and prediction, Folding problem, Sequence Analysis, Homology and Analogy.

Protein Information Resources:

Biological databases, Primary sequence databases, Protein Sequence databases, Secondary databases, Protein pattern databases, and Structure classification databases.

DNA sequence databases, specialized genomic resources

Unit-II **12 Hours**

Genome Information Resources: DNA Sequence Analysis:

Importance of DNA analysis, Gene structure and DNA sequences, Features of DNA sequence analysis, EST (Expressed Sequence Tag) searches, Gene hunting, Profile of a cell, EST analysis, Effects of EST data on DNA databases

Unit-III **12 Hours**

Pairwise Alignment Techniques:

Database searching, Alphabets and complexity, Algorithm and programs, Comparing two sequences, subsequence's, Identity and similarity, The Dot plot, Local and global similarity, different alignment techniques, Dynamic Programming, Pair wise database searching.

Multiple Sequence Alignment:

Definition and Goal, the consensus, computational complexity, Manual methods, Simultaneous methods, Progressive methods, Databases of Multiple alignments and searching

Unit-IV **12 Hours**

Secondary Database Searching:

Importance and need of secondary database searches, secondary database structure and building a sequence search protocol

Analysis Packages:

Analysis package structure, commercial databases, commercial software, comprehensive packages, packages specializing in DNA analysis, Intranet Packages, Internet Packages.

Text Books:

1. Introduction to Bioinformatics, by T K Attwood & D J Parry-Smith Addison Wesley Longman, 1999. ISBN: 0-582-32788-1
2. Bioinformatics a Beginner's GuidePaperback, by Jean Michael Claverie, CerdicNotredame**,** Wiley India Private Limited, 2003ISBN-13: 978-8126503803

Reference Books:

1. Introduction to Bioinformatics by Arthur M.Lesk, Third Edition, Oxford University Press, 2009

otal: 48 Hours

21CSH22 Design Patterns

4 0 0 4

Course Outcome:

At the end of the course students are able to:

1. Explain the catalogue of design patterns
2. Exemplify the appropriate pattern to solve object-oriented design problems using design patterns
3. Identify design solution using creational patterns
4. Identify structural patterns to solve design problems.
5. Design solutions by using behavioural patterns.
6. Choose a design pattern from the Pattern Community

COs–POs Mapping:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	3	-	3	-	-	-	-	3	-	-	2	-
CO2	2	2	2	-	2	-	-	-	-	2	-	-	2	-
CO3	2	3	3	-	1	-	-	-	-	2	-	-	2	-
CO4	2	2	2	-	2	-	-	-	-	2	-	-	2	-
CO5	2	3	2	-	3	-	-	-	-	2	-	-	2	-
CO6	2	3	3	-	2	-	-	-	-	2	-	-	2	-

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

UNIT – I **12 Hours**

Introduction: What Is a Design Pattern? Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

UNIT – II **12 Hours**

Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns. Structural Pattern Part-I: Adapter, Bridge, Composite.

UNIT-III **12 Hours**

Structural Pattern Part-II: Decorator, Façade, Flyweight, Proxy. Behavioral Patterns Part-I: Chain of Responsibility, Command, Interpreter, Iterator

UNIT-IV **12 Hours**

Behavioral Patterns Part-II: Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns. What to Expect from Design Patterns: A Brief History, the Pattern Community an Invitation, A Parting Thought.

Total: 48 Hours

TEXT BOOK:

1. Erich Gamma, Design Patterns, Pearson Education, 2015

REFERENCE BOOKS:

1. Mark Grand, Pattern's in JAVA Vol-I, Wiley Dream Tech, 2098
2. Mark Grand, JAVA Enterprise Design Patterns Vol-III, Wiley Dream Tech.2001

3. Eric Freeman, Head First Design Patterns, Oreilly-SPD, 2004
4. Alan Shalloway, Design Patterns Explained, Pearson Education, 2004

The Vision of GMRIT

- To be among the most preferred institutions for engineering and technological education in the country
- 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 very best
- An institution where ethics, excellence and excitement will be the work religion, while research, innovation and impact, the work culture

The Mission of GMRIT

- To turnout disciplined and competent engineers with sound work and life ethics
- To implement outcome based education in an IT-enabled environment
- To encourage all-round rigor and instill a spirit of enquiry and critical thinking among students, faculty and staff
- To develop teaching, research and consulting environment in collaboration with industry and other institutions

Department Vision

To be a preferred department of learning for students and teachers alike, with a commitment towards Academics & Research, serving the students in an atmosphere of innovation, critical thinking and making them Industry ready.

Department Mission

M1: To provide adaptable education in a collaborative and innovative environment in skilling the graduates to solve real world problems in the field of Computer Science and Engineering

M2: To prepare the students as critical thinking professionals with multidisciplinary research orientation and Innovation

M3: To instil ethical values and nurture the graduates who will be able to contribute to society

Program Educational Objectives

PEO 1: Employ logical and analytical skills in solving complex real-world engineering problems in the areas of computer Science and allied fields..

PEO 2: Adaptable to emerging technologies with enhanced professional skills and ability towards continuous learning, facilitating higher studies and research.

PEO 3: Demonstrate professional ethics, leadership qualities and promote inclusive and collaborative growth with human values towards societal interest.

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 of solutions)

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 the process and service industries apart from the community utilities.

(Program Specific)

PSO2: Ability to design, develop and implement management systems, E-Commerce tools and Web Apps for product development. (Program Specific)

Department of Computer Science and Engineering

Minimum Credits to be earned: 160 (for Regular Students) 120 (for Lateral Entry Students)

First Semester (Pattern-I/Pattern-II)

No.	Course Code	Course	POs	L	T	P	Credits
1	23PYX01/ 23CYX01	Engineering Physics/Chemistry	1,2,7,10,12/1, 2,6,7,12	3/3	0	0	3/3
2	23MAX01/ 23MAX02	Linear Algebra& Calculus/Differential Equations and Vector calculus	1,2,3,4,12/1,2, 3,4,12	3/3	0	0	3/3
3	23BEX01/ 23BEX02	Basic Electrical and Electronics Engineering/ Basic Civil & Mechanical Engineering	1,3,6,7,9,12/1, 2,3,6,7,8,12	3/3	0	0	3/3
4	23BEX03	Introduction to Programming	1,2,3,12	3	0	0	3
5	23BEX04/ 23HSX01	Engineering Graphics/Communicative English	1,5,10,12/1,9, 10,11,12	2/2	0	2/0	3/2
6	23PYX02/ 23CYX03	Engineering Physics Lab/Chemistry Lab	4,6,9,11,12/1, 6,7,9,12	0	0	2/2	1/1
7	23BEX05/ 23BEX06	Electrical & Electronics Engineering workshop/Engineering Workshop	4,5,6,9,12/1,9, 12	0	0	3/3	1.5/1.5
8	23BEX07	Computer Programming Lab	2,3,4,12	0	0	3	1.5
9	23HSX11	-/ECA (Yoga / Sports)	-	-	-	-/1	-/0.5
10	23HSX12	-/CCA (NSS/NCC/Community Service)	-	-	-	-/1	-/0.5
11	23BEX08	IT Workshop/-	1,2,3,4,9,12	0	0	2/-	1/-
12	23HSX02	- /Communicative English Lab	1,9,10,11,12	0	0	-/2	-/1
			Total	**14/14**	**0**	**12/12**	**20/20**

Second Semester(Pattern-I/Pattern-II)

1	23HSX01/ 23BEX04	Communicative English/ Engineering Graphics	10,12/1,5,10	2/2	0	0/2	2/3
2	23MAX02/ 23MAX01	Differential Equations and Vector calculus/Linear Algebra & Calculus	1,2,3,4,12/1,2, 3,4,12/	3/3	0	0	3/3
3	23CYX01/ 23PYX01	Chemistry /Engineering Physics	1,2,6,7,12/1,2, 7,10,12	3/3	0	0	3/3
4	23BEX02/ 23BEX01	Basic Civil & Mechanical Engineering/ Basic Electrical and Electronics Engineering	1,2,3,6,7,8,12/ 1,3,6,7,9,12	3/3	0	0	3/3
5	23CS201	Data Structures (CSE, CSE-AI&DS, CSE-AI&ML, IT)	1,2,3,12,PSO1	3/3	0	0	3/3
6	23CYX03/ 23PYX02	Chemistry Lab / Engineering Physics Lab	1,6,7,9,12/4,6, 9,11,12	0	0	2/2	1/1
7	23BEX06/ 23BEX05	Engineering Workshop/Electrical & Electronics Engineering workshop	1,9,12/1,4,5	0	0	3/3	1.5/1.5
8	23BEX08	IT Workshop/-	1,2,3,4,9,12	0	0	2/-	1/-
9	23HSX02	Communicative English Lab/-	1,9,10,11,12	0	0	2/-	1/-
10	23CS202	Data Structures Lab (CSE, CSE-AI&DS, CSE-AI&ML, IT)	2,3,4,12,PSO1	0	0	3/3	1.5/1.5
11	23HSX11	-/ECA (Yoga/ Sports)	-	-	-	-/1	-/0.5
12	23HSX12	-/CCA (NSS/NCC/Community Service)	-	-	-	-/1	-/0.5
			Total	**14/14**	**0**	**12/12**	**20/20**

No.	Course Code	Course	POs	L	T	P	Credits
Third Semester							
1	23CS301	Problem solving using Python	1,2,3,5,12	3	-	2	4
2	23HSX10	Engineering Economics and Project Management	1,2,10,11,PSO2	3	-	-	3
3	23CS303	Design and Analysis of Algorithms	1,2,3,PO12,PSO1	3	-	-	3
4	23CS304	Digital Logic Design	1,2,3,4,5,PSO1	3	-	2	4
5	23CS305	Discrete Mathematical Structures	1,2,3,12,PSO1	3	-	-	3
6	23CS306	Object Oriented Programming with JAVA	1,2,3,PO12,PSO2	3	-	-	3
7	23CS307	Design and Analysis of Algorithms Lab	1,2,3,4,10,12,PSO1	-	-	3	1.5
8	23CS308	JAVA Lab	1,2,3,4,10,12,PSO2	-	-	3	1.5
9	23ESX01	Employability Skills I	1,2,3,4,5,9,10,11,12	0	-	2	-
			Total	**18**	**-**	**12**	**23**
Fourth Semester							
1	23IT304	Database Management Systems	1,2,3,10,12,PSO2	3	-	-	3
2	23IT403	Operating Systems	1,2,12,PSO1,PSO2	3	-	-	3
3	23CS403	Computer Organization and Architecture	1,2,3,12,PSO1	3	-	-	3
4	23MA404	Probability and Statistics using Python	1,2,4,10,12	3	-	2	4
5	23CS405	Web Coding and Development	1,2, 3, 5, 8, PSO2	3	-	-	3
6	23IT308	Database Management Systems Lab	1,2,3,8,10,12,PSO2	-	-	3	1.5
7	23CS407	Web Coding and Development Lab	1,2,3,4,5,8,10,PSO2	-	-	3	1.5
8	23ESX01	Employability Skills I	1,2,3,4,5,9,10,11,12	0	-	2	2
			Total	**15**	**-**	**10**	**21**
Fifth Semester							
1	23EC502	Microprocessors and Microcontroller Programming (Integrated)	1,2,3,4,5	3	-	2	4
2	23CS502	Artificial Intelligence and Machine Learning	1,2,3,4,PSO1,PSO2	3	-	-	3
3	23CS503	Computer Networks (Integrated)	1,2,4,5,PSO1	3	-	2	4
4	23CS504	Theory of Computation	1, 2,3,PSO1	3	-	-	3
5		Elective I (Professional Elective)		3	-	-	3
6		Elective II (Open Elective I)		3	-	-	3
7	23CS507	Artificial Intelligence and Machine Learning Lab	1,4,5,12,PSO1,PSO2	-	-	3	1.5
8	23TPX01	Term Paper	1,4,10,12	-	-	3	1.5
9	23ESX02	Employability Skills II	1,2,3,58,10,12	0	-	2	-
10	23SIX01	Summer Internship I	1,2,8,10,12	-	-	-	1
			Total	**18**	**-**	**12**	**24**
Sixth Semester							
1	23CS601	Compiler Design	1, 2,12,PSO1	3	-	-	3
2	23CS602	Cryptography and Network Security	3, 6,8,12	3	-	-	3
3	23CS603	Software Engineering	1,2,3,5,11,PSO1	3	-	-	3
4		Elective III (Professional Elective)		3	-	2	4
5		Elective IV (Open Elective II)		3	-	-	3
6	23CS606	Case Tools Lab	3,4,5,11,PSO1	-	-	3	1.5
7	23MPX01	Mini Project	All POs & PSOs	-	-	3	1.5
8	23ESX02	Employability Skills II	1,2,5,8,10,12	0	-	2	2
9	23ATX01	Environmental Studies	1,7	-	-	-	-
10	23ATX02	Professional Ethics and Human Values	----	-	-	-	-

11	23ATX---	Audit Course	----	-	-	-	-
			Total	**15**	**-**	**10**	**21**
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	23SIX02	Summer Internship II	1,2,5,6,10,12	-	-	-	1
5	23PWX01	Project	All POs& PSOs	-	-	16	8
			Total	**9**	**-**	**16**	**18**
Eighth Semester							
1		Elective VIII (Professional Elective)		-	-	-	3
2		Elective IX (Open Elective IV)		-	-	-	2
3	23FIX01	Full Semester Internship (FSI)	1,2,5,8,9,10, PSO1,PSO2	-	-	-	8
			Total	**-**	**-**	**-**	**13**

List of Electives

Elective I							
Career Path I ,II,III and other Core Electives							
1	23CSC11	Neural Networks (AI&ML)	1,2,4,5,12	3	-	-	3
2	23CSC21	Backend Programming Languages (Full Stack Developer)	1,2,3,5,PSO1,PSO2	3	-	-	3
3	23ITC31	Fundamentals of Security (Cyber Security)	1,2,3,4,PSO1,PSO2	3	-	-	3
4	23CS004	Principles of Programming Languages	1, 2, 3, 4	3	-	-	3
5	23CS005	Mobile Computing	1,3, 8	3	-	-	3
6	23CS006	Distributed Operating Systems	1,2,3,12	3	-	-	3
Elective II							
Open Electives							
1	23OEX11	Data Wrangling and Preprocessing	1,2,3,5,12,PSO1	3	-	-	3
2	23OEX21	Fundamentals of Cloud Computing	2,6,7,8	3	-	-	3
Elective III							
Career Path I ,II,III and other Core electives							
1	23CSC12	Deep Learning (AI&ML)	1,2,3,4,5,PSO1	3	-	2	4
2	23CSC22	Web Application Framework (Full Stack Developer)	1,2,3,5,PSO1, PSO2	3	-	2	4
3	23ITC32	Cybernet Security (Cyber Security)	1,3,4,12,PSO1,PSO2	3	-	2	4
4	23CS007	Cloud Computing Essentials	2,5,6,7,8	3	-	2	4
5	23CS008	Internet of Things (Integrated)	1,3,4,7,PSO1	3	-	2	4
Elective IV							
Open Electives							
1	23OEX12	Data Visualization Techniques	1,2,3,PSO1,PSO2	3	-	-	3
2	23OEX22	Cloud Services using AWS	2,3,4,5,12	3	-	-	3
Elective V							
Career Path I ,II,III and other Core electives							
1	23CSC13	Natural Language Processing (AI&ML)	2,3,4,12,PSO1	3	-	-	3
2	23CSC23	Web Application Databases (Full Stack Developer)	1,3,5,PSO1	3	-	-	3
3	23ITC33	Cloud Security (Cyber Security)	1,2,4,8,12,PSO2	3	-	-	3
4	23IT010	Social Network Analysis	1,2,7,12	3	-	-	3
5	23CS011	Optimization Techniques	2, 3, 5,12	3	-	-	3
6	23CS012	Wireless Adhoc Networks	1,2,3,6,7,10	3	-	-	3
Elective VI: Professional Elective							
1	23CS014	Green Computing	2,3,6,7	3	-	-	3

2	23CS015	Software Project Management	1,2,3,11,12,PSO2	3	-	-	3
3	23CS016	Soft Computing	1,2,3,4	3	-	-	3
Elective VII:							
Open Electives							
1	23OEX13	Time series Analysis and Forecasting	1,2,3,PSO1	3	-	-	3
2	23OEX23	Cloud Security Essentials	1,2,3,PSO2	3	-	-	3
Elective VIII: Professional Elective							
1	23CS017	Fundamentals of Social Network Analysis	2, 4,5	-	-	-	3
2	23CS018	Information Retrieval Systems	1,2,3,4	-	-	-	3
3	23CS019	Fundamentals of Devops	1,3, 5,8,10	-	-	-	3
Elective IX:							
Open Electives							
1	23OEX14	AI for Business Intelligence	1,2,4,5,PSO1	2	-	-	2
2	23OEX24	Cloud-Based DevOps	1,2,3,4,5,PSO1	2	-	-	2

Audit Course							
1	23AT001	Communication Etiquette in Workplaces	-	-	-	-	-
2	23AT002	Contemporary India: Economy, Policy and Society	-	-	-	-	-
3	23AT003	Design The Thinking	-	-	-	-	-
4	23AT004	Ethics and Integrity	-	-	-	-	-
5	23AT005	Indian Heritage and Culture	-	-	-	-	-
6	23AT006	Intellectual Property Rights and Patents	-	-	-	-	-
7	23AT007	Introduction to Journalism	-	-	-	-	-
8	23AT008	Mass Media Communication	-	-	-	-	-
9	23AT009	Science, Technology and Development	-	-	-	-	-
10	23AT010	Social Responsibility	-	-	-	-	-
11	23AT011	The Art of Photography and Film Making	-	-	-	-	-
12	23AT012	Gender Equality for Sustainability	-	-	-	-	-
13	23AT013	Women in Leadership	-	-	-	-	-
14	23AT014	Introduction to Research Methodology	-	-	-	-	-
15	23AT015	Climate Change and Circular Economy	-	-	-	-	-

B. Tech. (Honors)							
Domain I (Data Engineering)							
1	23CSH11	Advanced Data Structures	2,3,4	4	-	-	4
2	23CSH12	Advanced Databases	2,3,4	4	-	-	4
3	23CSH13	Programming, Data Structures and Algorithms Using Python	2,3,4,5	4	-	-	4
4	23CSH14	Bioinformatics	2,3	4	-	-	4
5		MOOCs		4	-	-	4
Domain II (Modern Software Engineering)							
1	23CSH21	DevOps	1,3,5,8,10	4	-	-	4
2	23CSH22	Design Patterns	2,3	4	-	-	4
3	23CSH23	Advanced Software Engineering	1,3,4, PSO1	4	-	-	4
4	23CSH24	Robotic Process Automation	3,5, 8, PSO2	4	-	-	4
5		MOOCs		4	-	-	4
Domain III (Security)							
1	23CSH31	Computer Systems Security	1,2	4	-	-	4
2	23CSH32	Python Programming for Security	2,3,4	4	-	-	4
3	23CSH33	Management of Information Security	3,6,7	4	-	-	4
4	23CSH34	Computer Forensics	2,3	4	-	-	4
5		MOOCs		4	-	-	4
Domain IV (User Interface Design)							
1	23CSH41	Computer Graphics	1,2,3,4	4	-	-	4
2	23CSH42	Multimedia Systems	3,4	4	-	-	4

3	23CSH43	Human Computer Interaction	2,3	4	-	-	4
4	23CSH44	Mobile Programming	3,4	4	-	-	4
5		MOOCs		4	-	-	4

B. Tech. (Minors)							
Computer Science and Engineering							
1	23CSM01	Data Structures and Algorithms Using Python	--	4	-	-	4
2	23CSM02	Fundamentals of Data Base Management System	--	4	-	-	4
3	23CSM03	Fundamentals of Computer Networks & Operating Systems	--	4	-	-	4
4	23CSM04	Software Engineering	--	4	-	-	4
5	23CSM05	Fundamentals of Web Development(MOOCs)	--	4	-	-	4

3rd Semester

23CS301 Problem Solving using Python

3 0 2 4

Course Outcomes

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

1. Explain the fundamentals of Python with syntax and semantics.
2. Apply conditional and control flow statements, and analyze strings and regular expressions for pattern matching and text processing.
3. Apply the built-in data structures such as dictionaries, sets, lists, and tuples for data manipulation.
4. Apply the concepts of functions and functional programming techniques.
5. Apply the file Handling and exception handling techniques.
6. Describe the principles of object oriented programing in Python and demonstrate the use of Modules and Packages.

CO – PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	3	-	-	-	-	-	-	1	-	-
CO2	2	3	3	-	3	-	-	-	-	-	-	2	-	-
CO3	2	3	3	-	3	-	-	-	-	-	-	2	-	-
CO4	2	2	3	-	3	-	-	-	-	-	-	2	-	-
CO5	2	3	2	-	3	-	-	-	-	-	-	2	-	-
CO6	3	2	3	-	3	-	-	-	-	-	-	2	-	-

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

Unit I **12 Hours**

Introduction: History and need for python programming, Identifiers, Statements, Variables, Keywords, Input-Output statements, Indentation, command line arguments.

Data Types - Integers, Strings, Booleans. **Operators** - Arithmetic, Comparison/Relational, Assignment, Logical, Bitwise, Membership, Identity. **Expressions and order of evaluations -** Precedence and Associativity. **Built-in Functions** – Importing from Packages. **Conditional Statements**: If, Elif, Else statements, iteration statements - While, For - Break, Continue and pass statements.

Applications of Python, REPL (Read, Evaluation, Print, Loop)

Practical Components

1. Basic Operations: Validating and Converting Data Formats and Numeric Systems in Python
2. Control Statements: Implementing Fibonacci Series and Advanced List Operations
3. Array Manipulations : Matrix Operations: Transpose, Symmetry Check, and Multiplication

Unit II **12+6 Hours**

Collections: Lists: Create, Access, Slicing, Negative indices, List methods, List comprehensions – Tuples: Create, Indexing and slicing, Operations on tuples – Dictionary: Create, add, and replace values, Operations on dictionaries – Sets: Creation and operations.

Strings: Strings: Comparison, Formatting, Slicing, Splitting, Stripping . **Regular Expressions**: Matching, Search and replace, Patterns.

collections.deque, grouping with Dictionaries

Practical Components

1. Array Manipulations: Intersection, Kth Largest Element, and Maximum Subarray in Lists
2. Working with Dictionaries: Word Frequency Counting, Dictionary Merging, and Filtering
3. String Manipulation: Acronyms, Valid Parentheses, Anagram Check, and Common Prefix
4. Regular Expressions for Email Validation, Phone Number Extraction, and Password Strength Checking

Unit III **12 +8 Hours**

Functions- Defining Functions, Calling Functions, Types of Arguments, Recursion, Anonymous Functions: Lambda Expressions, Global and Local Variables. Functional Programming: map, filter and reduce, decorators. **Modules and Packages-** Built-in modules, User-Defined modules, itertools module. **Numpy and Pandas:** Introduction to Numpy and Pandas packages of python, Basic operations on Numpy and Pandas.

Higher order function, keyworded variable length arguments.

Practical Components

1. Recursive Functions: Prime Factorization, Subsets Generation, and Fibonacci Sequence
2. Exploring Default and Keyword Arguments, Variable Scopes, and Lambda Functions
3. Functional Programming: Map, Filter, and Reduce Operations
4. Itertools Applications: Counting, Cycling, Permutations, Combinations, and Sliding Windows
5. Working with Numpy Arrays and Pandas DataFrames

Unit IV **12+10 Hours**

Classes and Objects: Defining and instantiating Class, Class Attributes and Methods, Constructor, self-keyword - inheritance – overriding methods from parent class – Instances as Attributes, inner class. **File Handling-**Creating files, operations on files - Open, Close, Read and Write - Working with File Paths - File and Directory Operations. **Exception Handling-** Exception, try except block, Raising Exceptions, User Defined Exception.

polymorphism, Abstract class and abstract methods.

Practical Components

1. Object-Oriented Programming: Creating and Using Classes and Objects
2. Inheritance and Method Overloading/Overriding in Python
3. File Handling and Simple File Explorer Implementation
4. Exception Handling: Custom Exceptions

Total: 48+32 Hours

Textbook (s)

1. Richard L. Halterman, "Fundamentals of Python Programming" , 3rd Edition, Southern Adventist University, 2019.
2. Matthes, E. ,"Python Crash Course: A Hands-On, Project-Based Introduction to Programming", 2nd Edition, No Starch Press, 2019.
3. Willaim Mckenny, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" 2nd Edition, O'Reily 2017.

Reference(s)

1. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education India, 2015.
2. Mertz, D., "Functional Programming in Python", 1st Edition, O'Reilly Media,2015.
3. Kenneth A. Lambert. "Fundamentals of Python: First Programs", 2nd Edition, Publisher: Cengage Learning 2018
4. Python Programming: A Modern Approach, VamsiKurama, Pearson 2017 Learning Python, Mark Lutz, Orielly, 5th Edition.

23HSX10 Engineering Economics and Project Management

3 0 0 3

Course Outcomes

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

1. Illustrate the basic principles of engineering economics.
2. Demonstrate Cost-Volume-Profit (CVP) analysis in business decision making.
3. Implement the simple financial statements for measuring financial performance of a firm.
4. Evaluate investment proposals through various capital budgeting methods.
5. State key issues of organization, management and administration.
6. Determine the accurate project cost estimates and plan future activities.

COs – POs Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	-	-	-	-	-	-	-	2	1	-	-	2
CO2	3	2	-	-	-	-	-	-	-	1	2	-	-	2
CO3	1	3	-	-	-	-	-	-	-	3	2	-	-	2
CO4	1	3	-	-	-	-	-	-	-	3	2	-	-	2
CO5	2	2	-	-	-	-	-	-	-	1	3	-	-	2
CO6	1	2	-	-	-	-	-	-	-	2	3	-	-	2

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

Unit-I **11 Hours**

Introduction to Engineering Economics - Demand Forecasting & Cost Analysis

Concept of Engineering Economics – Types of efficiency – Managerial Economics Nature and Scope – Law of

Demand – Types of Elasticity of demand.

Demand Forecasting & Cost Analysis: Demand Forecasting: Meaning, Factors Governing Demand Forecasting, Methods of Demand Forecasting (Survey and Statistical Methods) – Cost Analysis: Basic Cost Concepts, Break Even Analysis.

Factors affecting the elasticity of demand – Supply and law of Supply

Unit-II **13 Hours**

Market Structures - Financial Statements & Ratio Analysis

Different type of Markets Structures – Features – Price Out-put determination under Perfect Competition and Monopoly

Financial Statements & Ratio Analysis: Introduction to Financial Accounting – Double entry system – Journal – Ledger – Trail Balance – Final Accounts (with simple adjustments) – Financial Analysis through Ratios: Interpretation of Liquidity Ratios (Current Ratio and quick ratio), Activity Ratios (Inventory turnover ratio and Debtor Turnover ratio, Creditors Turnover Ratio, Capital Turnover Ratio), Solvency Ratios (Debt- Equity ratio, Interest Coverage ratio), and Profitability ratios (Gross Profit Ratio, Net Profit ratio, Operating Ratio, P/E Ratio and EPS).

Price output determination under Monopolistic markets, Accounting concepts and conventions

Unit-III **12 Hours**

Investment Decisions and Fundamentals of Management

Time Value of Money – Capital Budgeting: Meaning, Need and Techniques of Capital Budgeting

Introduction to Management: Nature – Importance – Classical Theories of Management: F.W.Taylor's and Henri Fayol's Theory – Functions and Levels of Management – Decision Making Process – Inventory Control, Objectives, Functions – Analysis of Inventory – EOQ.

Maslow & Douglas McGregor theories of Management, ABC Analysis

Unit-IV **12 Hours**

Project Management

Introduction – Project Life Cycle and its Phases – Project Selection Methods and Criteria – Technical Feasibility – Project Control and Scheduling through Networks – Probabilistic Models of Networks – Time-Cost Relationship (Crashing) – Human Aspects in Project Management: Form of Project Organization – Role & Traits of Project Manager.

Sources of Long-term and Short-term Project Finance

Total: 48 Hours

Textbook (s)

Pravin Kumar, Fundamentals of Engineering Economics, Wiley India Pvt. Ltd. New Delhi, 2015
Rajeev M Gupta, Project Management, 2ndEd., PHI Learning Pvt. Ltd. New Delhi, 2014
PanneerSelvam. R, Engineering economics, 2nd Ed., Prentice Hall of India, New Delhi, 2013

Reference (s)

R.B.Khanna, Project Management, PHI Learning Pvt. Ltd. New Delhi, 2011
R. PanneerSelvam&P.Senthil Kumar, Project Management, PHI Learning Pvt. Ltd. New Delhi, 2010
A. Aryasri, Management Science, 4thEd., Tata McGraw Hill, 2014
A. Aryasri, Managerial Economics and Financial Analysis, 4thEd., Tata McGraw Hill, 2014
Koontz &Weihrich, Essentials of Management, 6thEd., TMH, 2010
Chuck Williams and Mukherjee, Principle of Management 7thEd., Cengage Learning, 2013

23CS303 Design and Analysis of Algorithms

3 0 0 3

Course Outcomes

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

1. Analyse the performance of algorithms using space and time complexity, and asymptotic notations.
2. Implement algorithms using brute force and divide and conquer approaches.
3. Apply advanced data structures and graph algorithms to solve complex computational problems effectively.
4. Demonstrate the ability to solve problems efficiently using greedy approach.
5. Develop problem-solving skills by tackling complex computational problems using dynamic programming and backtracking.
6. Apply branch and bound techniques to solve complex problems, and understand the concepts of NP-completeness to identify NP-complete problems.

COs–POs Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	-	-	-	-	-	-	-	-	1	3	-
CO2	2	2	3	-	-	-	-	-	-	-	-	1	3	-
CO3	3	3	3	-	-	-	-	-	-	-	-	2	3	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	3	-
CO5	3	2	2	-	-	-	-	-	-	-	-	2	3	-
CO6	3	3	2	-	-	-	-	-	-	-	-	2	3	-

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

Unit I **12 hours**

Algorithm Analysis and Divide and Conquer

Fundamentals of algorithmic problem solving - Analysis framework - Performance Analysis: Space and Time complexity - Growth of Functions: Asymptotic Notations - Mathematical Analysis of Non-recursive and Recursive algorithms - Master's theorem - Brute Force String Matching Algorithm- Divide and Conquer: Merge sort, Quick sort, Fibonacci search.

Interpolation search and Radix sort

Unit II **12 hours**

Advanced Data Structures and Graph Algorithms:

Balanced Search Trees: AVL trees, B-trees - Heaps: Heapsort - Disjoint set: Find and Union Operations - Graphs: Introduction, Representation - Graph Traversals: BFS, DFS, Topological sort.

Tries and Strongly Connected Components

Unit III **12 hours**

Greedy and Dynamic Programming

Greedy Method: General method, Job sequencing with deadlines, Optimal Merge Patterns - Minimum cost spanning trees: Prim's and Kruskal's Algorithms - Single source shortest path problem: Dijkstra's Algorithm - Dynamic Programming: General method, Matrix chain multiplication, 0/1 knapsack problem, All pairs shortest path problem: Floyd Warshall Algorithm.

Graph coloring problem, Huffman Trees and Codes

Unit IV **12 hours**

Backtracking and Brach and Bound

Backtracking: General method, n-queen problem, sum of subsets problem, Hamiltonian Circuit problem. Branch and Bound: General method, Knapsack Problem, Traveling sales person problem, Assignment Problem. Complexity Classes : P, NP and NP-Complete Problems.

Exhaustive Search, Space and Time Trade-Offs

Total: 48hours

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./ PearsonEducation
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 BaughandM arcus Schaefer,Pearson Education.
3. www.geeksforgeeks.org
4. www.hackerearth.com
5. www.tutorialspoint.com

23CS304 Digital Logic Design

3 0 2 4

Course Outcomes

1. Demonstrate 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. Analyze Sequential circuits like latches and flip-flops.
6. Analyze and Design sequential circuits like Registers and Counters.

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	-	-	-	-	-	-	-	-	-	-	2	-
CO2	3	3	1	1	1	-	-	-	-	-	-	-	2	-
CO3	3	1	3	2	2	-	-	-	-	-	-	-	3	-
CO4	2	3	3	2	2	-	-	-	-	-	-	-	3	-
CO5	3	2	2	-	2	-	-	-	-	-	-	-	3	-
CO6	3	3	3	2	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

Realization of Logic gates and verification of Truth tables
Realization of basic gates using Universal gates (NOR)
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.

23CS305 Discrete Mathematical Structures

3 0 0 3

Course Outcomes

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

1. Construct and Verify the Correctness of Statements using Propositional and Predicate Logic.
2. Illustrate the Operations on Discrete Structures such as Relations and Functions
3. Demonstrate the Lattices and Algebraic Structures for the Modeling of Objects
4. Utilize the Counting Techniques to Solve Combinatorics Problems
5. Make use of the Binomial and multinomial Theorems to Solve Problems involving Recurrence Relations and Generating Functions
6. Demonstrate Graphs and Trees as Tools to Visualize and Simplify Situations

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	-	-	-	-	-	-	-	-	1	1	-
CO2	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO3	3	1	1	-	-	-	-	-	-	-	-	1	1	-
CO4	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO5	3	2	2	-	-	-	-	-	-	-	-	1	1	-
CO6	3	3	3	-	-	-	-	-	-	-	-	1	1	-

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

Unit I **12 Hours**

Statement Logic and Predicate 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

Program Correctness - Design of Logic Circuits

Unit II **12 Hours**

Binary Relations, Functions and Algebraic Structures

Binary Relations - Properties and Operations - Relational Graphs - Relation Matrices - Equivalence Relations -Compatibility Relations - Partial Ordering Relations - HasseDiagram
Lattices - Properties and Types
Functions – Types of functions - Inverse Function - Composition of Functions
Algebraic Structures – Properties - Semi Groups – Monoids – Groups - Abelian Groups – Subgroups

Relations in Databases – Rings - Fields

Unit III **12 Hours**

Combinatorics, Generating Functions and Recurrence Relations

Basics of Counting – Cardinality - Sum and Product Rules - Permutations and Combinations with Repetitions Binomial Coefficients - Binomial and Multinomial Theorems - The Principle of Inclusion and Exclusion - Pigeonhole Principle.
Generating Functions - Function of Sequences - Calculating Coefficient of Generating Functions - Recurrence Relations - Solving Recurrence Relation by Substitution.

Algorithm Analysis - Time and Space Complexity - Combinatorics for Bioinformatics

Unit IV **12 Hours**

Basics of Graph Theory

RepresentationofGraphs-DegreeofaGraph-HandshakingProperty-CompleteGraphs-RegularGraphs -Bipartite Graphs -Walk – Path – Circuit – Cycle - Planar Graphs - Euler's Formula – Isomorphism - Euler Circuit and Hamilton Cycle - Chromatic Number - Trees and its Properties - Binary Trees and its Types.

Operations on Graphs, Four Color Problem, Applications of Graphs and Trees, Ternary Tree

Total: 48 Hours

Textbook(s)

1. Kenneth H. Rosen, Discrete Mathematics and Applications, 7thEdition, Tata McGraw Hill,2015.
2. J.L.Mott,A.Kandel&T.P.Baker,DiscreteMathematicsforComputerScientists&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,2097.
2. C. L. Liu and D. P. Mohapatra, Elements of Discrete Mathematics, A Computer Oriented Approach, 3rdEdition, Tata McGraw Hill,2008.
3. D. S. Chandrasekharaiah, Mathematical Foundation to Computer Science, Prism Books Pvt. Ltd, Hyderabad, 4thEdition,2012.
4. Ralph P. Grimaldi, B. V. Ramana, ―Discrete and Combinatorial Mathematics - An Applied Introduction‖, Pearson Education, India, 5th Edition,2011.
5. Seymour Lipschutz and Marc Lipson, Discrete Mathematics, Revised 3rdEdition, Schaum's Outline Series, Tata McGrawHill, New Delhi,2009.

Web References:

1. http://www.web.stanford.edu/class/cs103x
2. http://www.cs.odu.edu/~cs381/cs381content/web_course.html
3. http://www.cse.iitd.ernet.in/~bagchi/courses/discrete-book
4. http://www.saylor.org/course/cs202/
5. http://www.nptel.ac.in/courses/106106094/

23CS306 Object Oriented Programming with JAVA

3 0 0 3

Course Outcomes

1. Apply the object oriented programming concepts to develop modular and reusable software solutions.
2. Develop applications using different types of inheritances
3. Implement user-defined packages to promote code reusability and modular programming.
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

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	-	-	-	-	-	-	-	-	2	-	1
CO2	3	3	1	-	-	-	-	-	-	-	-	2	-	1
CO3	3	3	1	-	-	-	-	-	-	-	-	2	-	1
CO4	2	2	2	-	-	-	-	-	-	-	-	2	-	2
CO5	3	2	2	-	-	-	-	-	-	-	-	2	-	2
CO6	3	2	2	-	-	-	-	-	-	-	-	2	-	2

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

Unit I **12 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 **12 Hours**

Inheritance, Packages & Interface

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

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

Packages: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, Member access rules.

Nested–Inner Class & Anonymous Classes–Generic Class Types

Unit III **11 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 **13 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

23CS307 Design and Analysis of Algorithms Lab

0 0 3 1.5

Course Outcomes

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

1. Analyze the performance of algorithms by implementing and measuring their space and time complexity using asymptotic notations.
2. Implement algorithms using brute force and divide and conquer approaches, and assess their efficiency.
3. Implement advanced data structures and graph algorithms to solve complex computational problems, and evaluate their performance.
4. Solve optimization problems efficiently using a greedy approach, and analyze the results.
5. Develop dynamic programming solutions to tackle complex computational problems, and measure their effectiveness.
6. Apply backtracking and branch and bound techniques to solve complex problems.

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	2	-	-	-	-	-	1	-	2	2	-
CO2	3	2	2	2	-	-	-	-	-	1	-	2	2	-
CO3	3	3	3	3	-	-	-	-	-	1	-	2	2	-
CO4	3	2	3	3	-	-	-	-	-	1	-	2	2	-
CO5	3	2	2	3	-	-	-	-	-	1	-	2	2	-
CO6	3	3	2	2	-	-	-	-	-	1	-	2	2	-

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

List of Experiments

1. Analysis of recursive algorithms
2. Analysis of Non-recursive algorithms
3. Solve the String matching problem using Brute Force approach
4. Solve sorting problem using divide and conquer approach
5. Implement heapsort algorithm using a max-heap.
6. Implement Simple Union and Find operation on disjoint-sets
7. Implement the graph traversal algorithm.
8. Implement topological sorting of a directed acyclic graph (DAG).
9. Find the minimum cost spanning tree for a given graph using greedy approach
10. Implement dijkstra's algorithm using greedy approach
11. Solve matrix chain multiplication problem using dynamic programming
12. Solve the 0/1 knapsack using dynamic programming
13. Implement Floyd algorithm using dynamic programming
14. Find the sum of subsets using backtracking technique
15. Solve N queen problem using the backtracking technique.
16. Find the solution for Assignment Problem using branch and bound method

Indicative list of Augmented Experiments:

1. Basic File System Simulation Using B-Trees
2. Implementation of Huffman Coding for Data Compression
3. Topological Sorting of Courses with Prerequisites
4. Implementation of Tries for Auto-Completion and Spell Checking
5. Route Optimization for a Delivery Service Using Traveling Salesman Problem (TSP) Algorithms
6. Implementation of a Priority Queue Using a Binary Heap for Task Scheduling
7. Graph Coloring Problem for Map Coloring
8. Dynamic Programming for Optimal Binary Search Tree Construction
9. Visualizing Strongly Connected Components in Social Network Analysis
10. Rod Cutting Problem

Reading Material (s)

1. Design and Analysis of Algorithms Lab manual, Department of CSE and IT, GMRIT
2. https://www.leetcode.com

23CS308 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. Design Graphical User Interface using AWT components
6. Design Graphical User Interface using Swing components

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	1	-	-	-	-	-	2	-	2	-	2
CO2	3	3	2	1	-	-	-	-	-	2	-	2	-	2
CO3	3	3	2	2	-	-	-	-	-	2	-	2	-	2
CO4	3	3	2	3	-	-	-	-	-	2	-	3	-	3
CO5	3	3	3	2	-	-	-	-	-	2	-	2	-	3
CO6	3	3	3	2	-	-	-	-	-	2	-	3	-	3

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

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 **calculatePermutations** 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 **calculateCombinations** 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

23ESX01 Employability Skills I

0 0 2 0

Course Outcomes

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Develop life skills with behavioral etiquettes and personal grooming.
3. Assess analytical and aptitude skills.
4. Develop algorithms for engineering applications.
5. Solve engineering problems using software.
6. Utilize simulation tools for testing

COs –POs Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	2	3	2				2	2	2	3		
CO2	3	2	3	2	3				2	2	2	3		
CO3	3	2	3	2	3				2	2	2	3		
CO4	3	2	3	2	2				3	3	3	3		
CO5	3	2	3	2	2				3	3	3	3		
CO6	2	1	2	3	2				3	3	3	3		

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

Unit-I

Communication Skills **8 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 **8hours**

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

Unit-II

Competitive Programming Phase – I **16 Hours**

Module-1:

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:

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

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

4th Semester

23IT304 Database Management Systems

3 0 0 3

Course Outcomes

1. Explain the fundamental concepts of database and Various data models
2. Ilustrate the Relational Algebra and integrity constraints for Query execution
3. Apply the relational database to formulate the SQL Commands
4. Make use of Normalization to improve database design and schema refinement
5. Analyze the concept of transaction management, recovery and serializability in databases
6. Classify the concurrency control protocols, different types of data and Indexing techniques

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	1	-	-	-	-	-	-	-	-	2	-	2
CO2	3	2	3	-	-	-	-	-	-	2	-	2	-	2
CO3	3	3	3	-	-	-	-	-	-	2	-	2	-	2
CO4	3	3	3	-	-	-	-	-	-	2	-	2	-	3
CO5	3	2	3	-	-	-	-	-	-	2	-	2	-	3
CO6	2	1	2	-	-	-	-	-	-	2	-	2	-	3

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

Unit I **12 Hours**

Introduction to DBMS and ERModel

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 **12 Hours**

Introduction to Relational Model and Basic SQLQueries

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, OuterJoins.

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, Database Systems design, Implementation and Management, 9thEdition, 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, 8thEdition,2006

23IT403 Operating Systems

3 0 0 3

Course Outcomes

1. Explain the different services provided by operating system at different level.
2. Analyze various CPU Scheduling Algorithms and synchronization techniques.
3. Identify deadlock prevention and avoidance measures.
4. Compare and contrast main memory and virtual memory.
5. Illustrate the functionality of file systems.
6. Evaluation of various disk scheduling algorithms.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	-	-	-	-	-	-	-	-	2	-	2
CO2	3	2	-	-	-	-	-	-	-	-	-	2	2	3
CO3	3	3	-	-	-	-	-	-	-	-	-	2	2	3
CO4	3	3	-	-	-	-	-	-	-	-	-	2	2	2
CO5	2	2	-	-	-	-	-	-	-	-	-	2	2	2
CO6	2	2	-	-	-	-	-	-	-	-	-	2	2	2

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

Unit I **12 Hours**

Operating-Systems Overview and Process Management

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 **12 Hours**

Inter Process Communication Mechanism

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 **12 Hours**

Memory Management

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 **12 Hours**

File System Interface &I/O Systems

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.

23CS403 Computer Organization and Architecture

3 0 0 3

Course Outcomes

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

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	-	1	2	-
CO2	3	2	2	-	-	-	-	-	-	-	-	1	2	-
CO3	3	3	2	-	-	-	-	-	-	-	-	1	2	-
CO4	3	2	2	-	-	-	-	-	-	-	-	1	2	-
CO5	3	2	2	-	-	-	-	-	-	-	-	1	2	-
CO6	3	2	2	-	-	-	-	-	-	-	-	1	3	-

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

Unit I **13 Hours**

Overview & Micro operation

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 **11 Hours**

Arithmetic Operations

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 word Parallelism.

BCD Adder-BCD Subtraction

UNIT III **11 Hours**

Control Unit and Memory Systems

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 **13 Hours**

Parallelism

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

23MA404 - 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's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	-	1	-	-	-	-	-	2	-	2	-	-
CO2	3	2	-	3	-	-	-	-	-	3	-	2	-	-
CO3	3	2	-	3	-	-	-	-	-	3	-	1	-	-
CO4	3	1	-	3	-	-	-	-	-	3	-	3	-	-
CO5	3	1	-	2	-	-	-	-	-	3	-	1	-	-
CO6	3	1	-	3	-	-	-	-	-	3	-	3	-	-

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.

23CS405 Web Coding and Development

3 0 0 3

Course Outcomes

1. Design static web pages using HTML tags and CSS.
2. Describe the basic and advanced concepts of Java Scripts to design dynamic web pages.
3. Develop portable and dynamic web pages.
4. Demonstrate PHP server-side scripting language to develop business logic.
5. Make use of database connectivity to communicate database server from web server.
6. Develop Dynamic Web Pages using Flask.

COs–POs Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	-	-	-	-	2	-	-	-	-	-	3
CO2	3	3	3	-	3	-	-	2	-	-	-	-	-	3
CO3	2	2	3	-	2	-	-	2	-	-	-	-	-	3
CO4	2	2	3	-	2	-	-	2	-	-	-	-	-	3
CO5	2	2	1	-	3	-	-	2	-	-	-	-	-	3
CO6	2	3	3	-	3	-	-	2	-	-	-	-	-	3

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

Unit I **12 hours**

HTML Common tags: Lists, Tables, images, Forms, Links and Navigation.

CSS: Introduction, CSS Properties, Selectors, Combinators, Controlling Fonts, Forms, Pseudo classes, Navigation Bar, Layouts, The Box Model

Bootstrap

Responsive Design, Layouts, Grids, Media Queries, **Components**- Forms, Drop Downs, Cards, Input Groups, Collapse, List Groups, Navbar, Popovers, Flex Box.

Introduction to HTML5, XML, CSS3

Unit II **12 hours**

Java Script: Variables &Scope, Data types, Control Structures, Array Methods, Functions, Events, Validations, Objects, Document Object Model (DOM), Browser Object Model (BOM)

Advanced JavaScript: Arrow functions, Template Literals, Array Destructing, Object Destructing, Classes and Modules – ES6 imports and exports

Transpilation, Iterators and Generators

Unit III **12 hours**

PHP

Introduction, Model 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 Authentication.

PHP Cookies, PHP Exception Handling, Introduction to PHP laravel

Unit IV **12 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.

Text Book(s)

1. Programming the World Wide Web, 8th edition Robert W. Sebesta, Pearson.
2. Web Coding Development All In One for Dummies, Paul McFedries, Wiley.
3. Bootstrap: Responsive Web Development,1st Edition, jake spurlock foreword by davewiner, O'Reilly publications.
4. Advanced Java Script Speed up web development with powerful and benefits of Java Script, Zachary Shute, Packt Publishers.

Reference(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.

23IT308 Database Management Systems Lab

0 0 3 1.5

Course Outcomes

1. Demonstrate ER Modelling concepts to design the Database
2. Apply integrity constraints on a database
3. Apply different DDL, DML, DCL, TCL commands in creation and manipulation of Database
4. Implement subqueries to make the complex queries more readable and apply joins to combine different relations to get required data from database
5. Implementation of SQL conversion, number and string functions
6. Experiment with triggers, cursors and functions to maintain referential integrity of data

CO-PO Mapping

CO’s	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	2	-	-	-	2	-	2	-	2	-	3
CO2	3	2	2	3	-	-	-	2	-	2	-	2	-	2
CO3	3	2	2	2	-	-	-	2	-	2	-	2	-	3
CO4	3	2	3	2	-	-	-	2	-	2	-	2	-	3
CO5	3	2	2	3	-	-	-	2	-	2	-	2	-	3
CO6	3	2	2	2	-	-	-	2	-	2	-	2	-	2

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

List of Experiments

Experiment 1

Application: 1

"Established in 1977, "Roadway Travels" has been actively operating for decades, providing connectivity across various destinations in India through a fleet of several buses. Headquartered in Hyderabad, the company now aims to modernize its operations in the following key areas:

1. Reservation System:

Managing reservations is the responsibility of the booking office. Customers can reserve their seats up to 60 days in advance, utilizing either cash or credit payment methods. In instances where tickets are unavailable, customers receive wait-listed tickets, which can be confirmed once cancellations occur.

2. Ticketing:

The booking office handles ticketing processes, including sales and issuance. Both cash and credit payments are accepted for this purpose.

3. Cancellation and Modification:

Cancellation services are also facilitated at the booking office. Applicable cancellation charges will be levied as per company policy. Furthermore, modifications to reservations can be processed within this domain. Notably, wait-listed tickets that remain unconfirmed are eligible for full refunds."

The Following are the entities:
Bus (Busno, Source, Destination, coachtype)
Reservation (PNRNo,Busno, NoofSeats, JourneyDt, ContactNo, Address)
Ticket (TicketNo, BusNo, JourneyDt, Source, Destination, Dept_Time, Age, Sex
Passenger (TicketNo, PNRNo, Name, ContactNo, Age, Sex)
Cancellation (PNRNo, JourneyDt, SeatNo, ContactNo)

Application: 2

Online Learning Platform: An online learning platform requires an ER model to handle courses, students, instructors, and enrollments. Courses have titles, descriptions, and durations. Instructors have names, expertise, and contact details. Students have profiles with names, email addresses, and student IDs. Enrollments link students to courses, including enrollment dates.

Entities and Attributes:

Course (Course_ID (Primary Key), Title, Description, Duration (in weeks or hours)Level (Beginner, Intermediate, Advanced), Category (e.g., Programming, Mathematics, Language), Price, Language, Thumbnail Image URL,)

Instructor (Instructor_ID (Primary Key), Name, Bio, ContactInfo (Email, Phone), Expertise (Areas of specialization), ProfilePictureURL)

Student (Student_ID (Primary Key), FirstName. LastName, Email, DOB, Address, Contact)

Enrollment(Enrollment_ID (Primary Key), Course_ID (Foreign Key referencing Course), Student_ID (Foreign Key referencing Student), EnrollmentDate, Status (Enrolled, Completed, Dropped, etc.))

Lesson(Lesson_ID (Primary Key), Course_ID (Foreign Key referencing Course), Title
Order (Sequence within the course), Duration (in minutes or hours), Content (Text, Video URL, PDF URL, etc.))

Quiz(Quiz_ID (Primary Key), Course_ID (Foreign Key referencing Course), Title, Number of Questions, Passing Score)

Question(Question_ID (Primary Key), Quiz_ID (Foreign Key referencing Quiz), Question Text, Question Type (Multiple Choice, True/False, Short Answer, etc.))

Answer(Answer_ID (Primary Key), Question_ID (Foreign Key referencing Question), AnswerText, Correct_Answer_Indicator)

Application: 3

Music Streaming Service: A music streaming service needs an ER model to organize its content and user interactions. The system should handle songs, albums, artists, playlists, and users. Songs have titles, durations, and release years. Albums include album names, release dates, and cover art. Artists have names, genres, and biographies. Users have profiles with names, email addresses, and subscription details. Playlists contain songs and are created by users.

Application: 4

Company Database: The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations. A department controls several projects, each of which has a unique name, a unique number, and a single location. We store each employee's name, social security number, address, salary, sex, and birthdate. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's name, sex, birthdate, and relationship to the employee.

Entities are:
Company Database:
EMPLOYEE (SSN, EName, Address, Sex, Bdate, Salary, SuperSSN, DNo)
DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate)
DLOCATION (DNo,DLoc)
PROJECT (PNo, PName, PLocation, DNo)
WORKS_ON (SSN, PNo, Hours)
DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship)

Apply ER Model for the above applications and identify the relations present in each application represent them in ER diagram.& Convert ER model to Relational Model.

Note: Students need to form a group minimum of 3 students and maximum of 5, and should design at least two applications into ER model by using EdrawMax software and convert them into relational model

Experiment 2

Creating users, roles and grant privileges to users and relations.

Experiment 3

Create and alter tables for Company Database in SQL using Integrity Constraints.

Integrity Constraints are CHECK, NOT_NULL, DEFAULT, UNIQUE, PRIMARY KEY, FOREIGN KEY.
Use atleast one constraint in one relation to understand the usage of constraints on database.
Schema:
CREATE TABLE employee (Fname varchar (15) NOT NULL, Minit varchar(1), Lname varchar(15) NOT NULL, SSN varchar(9) NOT NULL, Bdate date, Address varchar(30), Sex varchar(1), Salary float, Super_ssn varchar(9), Dno int NOT NULL, PRIMARY KEY (ssn), FOREIGN KEY (Super_ssn) REFERENCES employee(ssn));

CREATE TABLE department (Dnamevarchar(15) NOT NULL, Dnumber int NOT NULL, Mgr_ssn varchar(9) NOT NULL, Mgr_start_date date, PRIMARY KEY (Dnumber), UNIQUE(Dname), FOREIGN KEY (Mgr_ssn) REFERENCES employee(ssn));

CREATE TABLE dept_locations (Dnumber int NOT NULL, Dlocationvarchar(15), PRIMARY KEY (Dnumber, Dlocation), FOREIGN KEY (Dnumber) REFERENCES department (Dnumber));

CREATE TABLE project (Pnamevarchar(15) NOT NULL, Pnumber int NOT NULL, Plocation varchar(15), Dnum int NOT NULL, PRIMARY KEY (Pnumber), UNIQUE (Pname), FOREIGN KEY (Dnum) REFERENCES department (Dnumber));

CREATE TABLE works_on (Essnvarchar(9) NOT NULL, Pno int NOT NULL, Hours float NOT NULL, PRIMARY KEY (Essn, Pno), FOREIGN KEY (Essn) REFERENCES employee(ssn), FOREIGN KEY (Pno) REFERENCES project(Pnumber));

CREATE TABLE dependent (Essnvarchar(9) NOT NULL, Dependent_name varchar(15) NOT NULL, Sex varchar(1), Bdate date, Relationship varchar(8), PRIMARY KEY (Essn, Dependent_name), FOREIGN KEY (Essn) REFERENCES employee(ssn));

Experiment 4

Execute basic SQL queries on Employee Database
To Create a database called Employee which consists of EMP, Dept, Salgrade, Project, Works_on and Dependent tables as relations in the database use Data Definition Language command (Create, Alter).
Run sql queries on database using DML commands like Insert, Update, Delete and for retrieval of data from database use Select statement.
Employee Database:
EMP (Empno, Ename, Job, Mgr, Hiredate, sal, comm, deptno)
DEPT (Deptno, Dname, Location)
SALGRADE(Grade, Losal, Hisal)
PROJECT (Pno, Pname, Plocation, Deptno)
WORKS_ON(Empno, Pno, Hours)
DEPENDENTS(EMPNO, DependentName, Sex, Bdate, Relation)

1.1 Insertion:
Insert rows into table by using datatypes like varchar, int, date, float etc.,
Syn: INSERT INTO employee (Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, Super_ssn, Dno) values (‘James’, ‘E’, ‘Borg’, 888665555, ’10-NOV-1937’, ‘430 Stone, Houston, TX’, ‘M’, 55000, NULL, 1);
1.2 Retrieval (Set operations)
Syn: Select Name from EMP Union Select Salary from EMP;
1.3 Updating
Update rows by changing the values in a row using update command syntax.
1.4 Deletion
Delete an entire row or multiple rows from the table using this command

Experiment 5

Implement the SQL operations on Employee Database created in Experiment 4.
Operations are: In, Any, All, Between, Like, Exists, Group by, Order by, having.

Employee Database:
EMP (Empno, Ename, Job, Mgr, Hiredate, sal, comm, deptno)

DEPT (Deptno, Dname, Location)
SALGRADE (Grade, Losal, Hisal)
PROJECT (Pno, Pname, Plocation, Deptno)
WORKS_ON (Empno, Pno, Hours)
DEPENDENTS (EMPNO, DependentName, Sex, Bdate, Relation)

Experiment 6

Implementing the concepts of Rollback – commit, checkpoints, and Views.

Rollback – commit, checkpoints:

Create Teacher table with the following fields(Name, DeptNo, Date of joining, DeptName, Location, Salary)

1. Insert five records
2. Give Increment of 25% salary for Mathematics Department .
3. Perform Rollback command
4. Give Increment of 15% salary for Commerce Department
5. Perform commit command.

Views:

To create different views, use the Emplpyee Database

1. Create a view to display Hiredate of all employees.
2. Create a view to display no of working hours of all employees with Plocation.

Experiment 7

Apply different types of joins availablein SQL on company database created in Experiment 4.

Types of Joins:

- Inner Join
- Outer Join
- Letf Join
- Right Join

Experiment 8

Apply subqueries and nested queries on company database (refer to experiment 4).

Experiment 9

Implement built in functions in oracle (Numeric, Aggregate functions) on Dual table(provided by Oracle) and Employee database.

Numeric Functions: ABS, CEIL, FLOOR, SQRT etc.,
Aggregate Functions: MIN, MAX, SUM, COUNT, AVG

Experiment 10

Implement String handling and Conversion functions on Dual table (provided by Oracle) and Employee database.

String Functions: Lowercase, Uppercase, Initcap, Rtrim, Ltrim, Lpad, Rpad etc.,
Conversion Functions: TO_CHAR(), TO_DATE(), TO_NUMBER().

Experiment 11

Sql Practice links from online platforms:

Employee Bonus	https://leetcode.com/problems/employee-bonus/
Big Countries	https://leetcode.com/problems/big-countries/
Employees Earning more than their managers	https://leetcode.com/problems/employees-earning-more-than-thei managers/
Managers with at least 5 direct reports	https://leetcode.com/problems/managers-with-at-least-5-direct-reports/
Second Highest Salary	https://leetcode.com/problems/second-highest-salary/
Count Salary Categories	https://leetcode.com/problems/count-salary-categories/

Experiment 12

Triangle	https://www.hackerrank.com/challenges/what-type-of triangle/problem?isFullScreen=true

Average Population of each country in a continent	https://www.hackerrank.com/challenges/average-population-of-eaccontinent/problem?isFullScreen=true
TopEarners	https://www.hackerrank.com/challenges/earnings-ofemployees/problem?isFullScreen=true
Occupations	https://www.hackerrank.com/challenges/occupations/problem?isFu
New Company	https://www.hackerrank.com/challenges/the-ompany/problem?isFu
15Days of Learning SQL	https://www.hackerrank.com/challenges/15-days-of-learning-sql/problem?isFullScreen=true

Experiment 13

Implement PL/SQL operations on Employee Database to retrieve one row as output.

Experiment 14

Implementing functions, stored procedures on company database using PL/SQL

Experiment 15

Implement cursors on Employee database using PL/SQL to retrieve multiple rows as output.

Experiment 16

Creating triggers using PL/SQL

E-RESOURCES AND OTHER DIGITAL MATERIAL

1. Prof Arnab Bhattacharya IIT Kanpur, SQL Introduction https://nptel.ac.in/courses/106104135/10
2. Prof Arnab Bhattacharya IIT Kanpur SQL: Updates, Joins, Views, and Triggers
3. https://nptel.ac.in/courses/106104135/11

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

23CS407 Web Coding and Development Lab

0 0 3 1.5

Course Outcomes

1. Design static web pages using HTML and CSS.
2. Design responsive web pages using Bootstrap for visually appealing web pages.
3. Create JavaScript functions for user authentication, form validation, dynamic content generation, and data visualization.
4. Develop business logic using PHP scripts.
5. Build database connectivity to facilitate communication between the Web Server and the Database Server.
6. Develop Web Applications using Flask framework.

CO–PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	3	1	2	-	-	1	-	2	-	-	-	3
CO2	2	3	3	3	2	-	-	3	-	2	-	-	-	3
CO3	2	2	2	3	2	-	-	1	-	2	-	-	-	3
CO4	2	2	3	3	2	-	-	3	-	2	-	-	-	3
CO5	2	2	3	3	3	-	-	2	-	2	-	-	-	3
CO6	2	2	2	3	2	-	-	2	-	2	-	-	-	3

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

List of Experiments

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. Apply different font styles, font families, font colors, animations, and other formatting styles to the above static web pages.

3. Apply CSS Box model to design above static web pages

4. Make the static pages Responsive and attractive using Bootstrap components (***Mobile View)

5. Write a JavaScript function that validates the student's login credentials. If the username and password are correct, redirect them to the test page; otherwise, show an error message.
6. Implement a form validation function for the registration page. Ensure that the password and confirm password fields match and that the email is in a valid format.
7. Write a JavaScript program that dynamically generates a set of multiple-choice questions and their answer choices from an array of question objects. Allow the user to select an answer and submit the test
8. a. Create a JavaScript function that calculates the user's test score based on their answers and displays it on the results page.
   b. Implement a chart or graph using a JavaScript library (e.g., Chart.js) to visualize the user's performance on the test, showing the number of correct and incorrect answers.

9. Create a PHP script that takes user registration data (e.g., username, email, password) from the registration form and inserts it into a database table for storing user information.

10. Develop a PHP script that validates the entered login credentials (username and password) against the database records. If valid, allow the user to log in.

11. Write a PHP script that receives the user's test answers from the test page and stores them in a database table along with the user's ID and timestamp.

12. Create a PHP script that retrieves the user's test results from the database and displays them on the results page. Include the user's score and a list of questions answered correctly and incorrectly.

13. Write a Flask application that establishes a connection to a SQLite database. Define a User model with fields like username, email, and password.

14. a. Create a Flask route for user registration. When a user submits the registration form, insert their information into the database.
    b. Implement Flask routes for student and admin login pages. Validate the entered login credentials against the database records.

15. Write a Flask route that retrieves a set of test questions from the database and displays them on the test page.

16. a. Develop a Flask route that receives the user's test answers and stores them in the database along with the user's ID and timestamp.
    b. Create a Flask route that retrieves the user's test results from the database and displays them on the results page.

List of Augmented Projects

1. WebCrawlers
2. Online Hospital Management
3. Online Shopping Project
4. Web based Appointment Systems
5. Scrapbook
6. Online Job Recruitment System
7. Attendance management system
8. Online Banking system
9. Online Exam
10. Railway Reservation system
11. Leave Application Management System
12. Address Book
13. Movie Rental Systems
14. Online Auction System
15. Student Information system

Reading Material (s)

1. Web Technologies Lab manual–Department of CSE & IT-GMRIT Rajam

23ESX01 Employability Skills I

0 0 2 2

Course Outcomes

1. Demonstrate oral communication and writing skills as an individual to present ideas coherently
2. Develop life skills with behavioral etiquettes and personal grooming.
3. Assess analytical and aptitude skills.
4. Develop algorithms for engineering applications.
5. Solve engineering problems using software.
6. Utilize simulation tools for testing

COs –POs Mapping

COs	PO1	PO2	PO5	PO8	PO10	PO12
CO1					3	2
CO2				1	2	2
CO3	2	1		2		
CO4	2		2			
CO5	2		2			
CO6	2		2			

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

Unit-I

Communication Skills **8 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 (Simple Topics).

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 **8 hours**

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

Unit-II

Competitive Programming Phase – II **16 Hours**

Module-1:

Fundamental mathematical concepts: Basic Mathematics, Number Theory, Geometry - Bit manipulation - Two-pointer technique.

Practice Problems:

Week	Topic	Practice Problem Links
1	Math-I	https://www.codechef.com/practice-old/topics/math
		https://leetcode.com/tag/math/
		https://www.codechef.com/practice-old/tags/mathematics
		https://www.interviewbit.com/courses/programming/math

2	Math-II	https://www.hackerrank.com/domains/mathematics
		https://www.codechef.com/practice-old/tags/mathematics
		https://www.hackerearth.com/practice/math/number-theory/basic-number-theory-1/practice-problems/
3	Number Theory	https://www.codechef.com/practice-old/tags/number-theory
		https://leetcode.com/tag/number-theory/
		https://www.hackerrank.com/domains/mathematics?filters%5Bsubdomains%5D%5B%5D=number-theory
4	Geometry	https://www.codechef.com/practice-old/tags/geometry
		https://www.hackerrank.com/domains/mathematics?filters%5Bsubdomains%5D%5B%5D=geometry
		https://leetcode.com/tag/geometry/
5	Bit Manipulation-I	https://www.codechef.com/practice-old/tags/bit-manipulation
		https://www.hackerearth.com/practice/basic-programming/bit-manipulation/basics-of-bit-manipulation/practice-problems/
6	Bit Manipulation-II	https://leetcode.com/tag/bit-manipulation/
		https://www.interviewbit.com/courses/programming/bit-manipulation
		https://www.hackerrank.com/domains/algorithms?filters%5Bsubdomains%5D%5B%5D=bit-manipulation
7	Two Pointer	https://www.interviewbit.com/practice/#topics[]=two-pointers
		https://leetcode.com/tag/two-pointers/
		https://www.codechef.com/practice-old/tags/two-pointers

Module-2:

Data Structures: Searching - Linked List - Stack – Queue – Graph Traversal Methods - Hashing Techniques.

8	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
9	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
10	Stacks	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
11	Queues	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

12	Linked List-I	https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=linked-lists
		https://leetcode.com/tag/linked-list/
13	Linked List-II	https://www.interviewbit.com/practice/#topics[]=linked-lists
		https://leetcode.com/tag/doubly-linked-list/
14	BFS	https://www.codechef.com/practice-old/tags/bfs&two-pointers
		https://leetcode.com/tag/breadth-first-search/
		https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/practice-problems/
15	DFS	https://leetcode.com/tag/depth-first-search/
		https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/practice-problems/
16	Hashing	https://www.codechef.com/practice-old/tags/hashing
		https://leetcode.com/problemset/?page=1&topicSlugs=hash-function

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