



**DEPARTMENT OF COMPUTER SCIENCE
KAZI NAZRUL UNIVERSITY**

ASANSOL 713 340

WEST BENGAL

www.knu.ac.in

Preamble:

The purpose of a Learning Outcome-based Curriculum Framework (LOCF) is to change the paradigm of higher education from a teacher-centric to learner-centric curriculum. It is hoped that this paradigmatic change will bring about a significant improvement in the quality of higher education and make the learners both competent and confident to face the challenges of a modern competitive world. The philosophy of this new curriculum framework is pragmatism, to realise that it is not enough for institutions of higher learning to produce good humans and responsible citizens of the country but also to produce employed graduates and postgraduates. After all, it is not prudent to expect an unemployed youth to cherish values like humanity and responsibility towards the nation; he/she first needs to have a productive employment to nourish such values.

LOCF seeks to make higher education in India learner-centric so that graduates and postgraduates not only have a more holistic understanding of their subject but also be able to better serve the humanity with dignity and honour, which can be expected only if they are able to secure productive employment after completing their higher education degrees.

Introduction to Learning Outcome Based Curriculum Framework (LOCF) in Kazi Nazrul University:

Two programs namely B.Tech. in Computer Science and Engineering (Data Science) [4 years Under-Graduate program] and MCA [2 years Post-Graduate program] in Kazi Nazrul University have been designed as a base for industry, research and application of knowledge. The syllabus and curricula of both the under graduate and the post graduate programmes have been developed following the UGC LOCF guidelines and through rigorous academic exercises after consulting eminent academic experts and feedback received from various stakeholders of the University. These two programs will enable the students to enhance their learning and to join the workforce in their respective fields. Kazi Nazrul University has an aim to develop the future generation learners sensitive towards the developmental challenges of the nation with special emphasis on the local developmental needs. The University also aims to foster this future generation of learners with a systematic understanding of global development need. The learning outcome-based curricula of different disciplines reflect the national as well as global sustainable needs listed below in the respective programme and course specific outcomes:

National needs:

- Promote Right to education
- Inculcate ethical and professional values
- Increase national and international visibility;

- leverage institutional strengths through strategic partnerships;
- enlarge the academic community within which to benchmark their activities;
- mobilise internal intellectual resources;
- add important, contemporary learning outcomes to student experience;
- Develop stronger research groups.
- Encourage multidisciplinary
- Promote Cross cultural exchanges
- Preservation of traditional knowledge
- Creating human resource for Economic growth
- Promotion of scientific mind-set and critical thinking

Sustainable development needs:

- Help to eradicate poverty
- Ensuring meal for all
- Promoting good health and well being
- Promoting quality education
- Promoting gender equality
- Initiatives for clean water and sanitization
- Programmes to reduce inequalities
- Develop sustainable cities and communities
- promote decent work and economic growth
- initiate industry-academia collaboration for innovative research
- encourage responsible consumer behaviour
- encourage pro-environment awareness

Program Outcomes (PO)s

The overall program outcome of the LOCF at PG level are to:

- help formulate postgraduate attributes, qualification descriptors, programme learning outcomes and course learning outcomes that are expected to be demonstrated by the holder of a Master's degree;
- enable prospective students, parents, employers and others to understand the nature and level of learning outcomes (knowledge, skills, attitudes and values) or attributes a graduate/postgraduate should be capable of demonstrating on successful completion of MSc
- maintain national standards and international comparability of standards to ensure global competitiveness, and to facilitate postgraduate mobility; and

- provide higher education institutions and their stake holders an important point of reference for setting and assessing standards.

Postgraduate Attributes

The postgraduate attributes reflect the particular quality and feature or characteristics of an individual, including the knowledge, skills, attitudes and values that are expected to be acquired by a postgraduate through studies at the higher education institution (HEI) such as a college or university. Such attributes include capabilities that help strengthen one's abilities for widening current knowledge base and skills, gaining new knowledge and skills, undertaking future studies and performing well in a chosen career and playing a constructive role as responsible citizen of the country. The Attributes define the characteristics of a student's university degree programme(s), and describe a set of characteristics/competencies that are designed to be transferable beyond the particular disciplinary area and programme contexts in which they have been developed. Such attributes are fostered through meaningful learning experiences made available through the curriculum, the total college/university experiences and a process of critical and reflective thinking.

The learning outcomes-based curriculum framework is based on the premise that every student is unique. Each student has his/her own characteristics in terms of previous learning levels and experiences, life experiences, learning styles and approaches to future career-related actions. The quality, depth and breadth of the learning experiences made available to the students while at the college/University help develop their characteristic attributes. The postgraduate attributes reflect both disciplinary knowledge and understanding and generic/global skills and competencies that all students in different academic fields of study should acquire/attain and demonstrate. Some of the desirable attributes which a postgraduate student should demonstrate will include the following:

- ***Disciplinary Knowledge:*** Demonstrate comprehensive knowledge and understanding of one or more disciplines that form a part of a programme of study, and knowledge and skills acquired from interaction with educators and peer group throughout the programme of study.
- ***Communication Skills:*** Express thoughts and ideas effectively in writing and orally, communicate with others using appropriate media, confidently share one's views and express herself/himself, demonstrate the ability to listen carefully, read and write analytically, and present complex information in a clear and concise manner to different groups.
- ***Critical Thinking:*** Apply analytic thought to a body of knowledge, analyse and evaluate evidence, arguments, claims, beliefs on the basis of empirical evidence, identify relevant assumptions or implications, formulate coherent arguments, critically evaluate practices, policies and theories by following scientific approach to knowledge development.

- ***Problem Solving:*** Demonstrate capacity to extrapolate from what one has learned and apply their competencies to solve different kinds of non-familiar problems, rather than replicate curriculum content knowledge and apply one's learning to real life situations.
- ***Analytical Reasoning:*** Demonstrate the ability to evaluate the reliability and relevance of evidence, identify logical flaws and holes in the arguments of others, analyse and synthesise data from a variety of sources, draw valid conclusions and support them with evidence and examples, and addressing opposing viewpoints.
- ***Research-related Skills:*** Demonstrate a sense of inquiry and capability for asking relevant/appropriate questions, problematising, synthesising and articulating, demonstrate the ability to recognise cause-and-effect relationships, define problems, formulate hypotheses, test hypotheses, analyse, interpret and draw conclusions from data, establish hypotheses, predict cause-and-effect relationships, plan, execute and report the results of an experiment or investigation.
- ***Collaboration/Cooperation/Team work:*** Demonstrate ability to work effectively and respectfully with diverse teams, facilitate cooperative or coordinated effort on the part of a group, and act together as a group or a team in the interests of a common cause and work efficiently as a member of a team.
- ***Scientific Reasoning using Quantitative/Qualitative Data:*** Demonstrate the ability to understand cause-and-effect relationships, define problems, apply scientific principles, analyse, interpret and draw conclusions from quantitative/qualitative data, and critically evaluate ideas, evidence and experiences from an open-minded and reasoned perspective.
- ***Reflective Thinking:*** Demonstrate critical sensibility to lived experiences, with self-awareness and reflexivity of both self and society.
- ***Information/Digital Literacy:*** Demonstrate capability to use ICT in a variety of learning situations, demonstrate ability to access, evaluate, and use a variety of relevant information sources and to use appropriate software for analysis of data.
- ***Self-Directed Learning:*** Demonstrate ability to work independently, identify appropriate resources required for a project, and manage a project through to completion.
- ***Multicultural Competence:*** Demonstrate knowledge of the values and beliefs of multiple cultures and a global perspective, effectively engage in a multicultural society, interact respectfully with diverse groups.
- ***Moral and Ethical Awareness/Reasoning:*** Demonstrate the ability to embrace moral/ethical values in conducting one's life, formulate a position/argument about an ethical issue from multiple perspectives, and use ethical practices in all work. Demonstrate the ability to identify ethical issues related to one's work, avoid unethical behaviour such as fabrication, falsification or misrepresentation of data or committing plagiarism, not adhering to intellectual property rights, appreciate environmental and sustainability issues, and adopt objective, unbiased and truthful actions in all aspects of work.

- **Community Engagement:** Demonstrate responsible behaviour and ability to engage in the intellectual life of the educational institution, and participate in community and civic affairs.
- **Leadership Readiness/Qualities:** Demonstrate capability for mapping out where one needs to go to "win" as a team or an organization, and set direction, formulate an inspiring vision, build a team who can help achieve the vision, motivate and inspire team members to engage with that vision, and use management skills to guide people to the right destination, in a smooth and efficient way.
- **Lifelong Learning:** Demonstrate the ability to acquire knowledge and skills, including 'learning how to learn' that are necessary for participating in learning activities throughout life, through self-paced and self-directed learning aimed at personal development, meeting economic, social and cultural objectives, and adapting to changing trades and demands of work place through knowledge/skill development/reskilling.

Department of Computer Science

Department of Computer Science, Kazi Nazrul University was established in 2020. Two Courses B.Tech. in Computer Science and Engineering (Data Science) [4 years Under-Graduate program] and MCA [2 years Post-Graduate program] are offered at present by the department. Two-year's MCA starts from 2022 academic year and four-year B.Tech. programme in Computer Science and Engineering (Data Science) was started in 2021 academic year. The thrust areas of the department are on Data Science, Artificial Intelligence, Machine learning, Internet-of-Things etc. The department is equipped with laboratory and seminar library. The Department of Computer Science aims to promote a balanced sense of theoretical concepts and its application in various fields among the students. Both, the students of these two courses are trained with professional ethics and scientific temper.

Courses offered in the Department

- *B.Tech. in Computer Science and Engineering (Data Science) [From 2021 onwards]*
- *MCA [From 2022 onwards]*

Mission

- ❖ *To translate the vision into reality, the Department is committed:*
- ❖ *To undertake high impact research in data analytics that has significant benefits for the individual, industry and society by enabling better decision making.*
- ❖ *Training and mentoring high calibre and work-ready analytics graduates, researchers and analytical experts.*

- ❖ *To promote entrepreneurs who are passionate about Data Science with a higher goal.*

Vision

The vision of the Department is to build up resources to meet the demand for Data Science professionals in demand for Data Science services in various fields. The department is focused to be a leader in applied statistical sciences, provide consultancy and perform research for overall development of society.

Structure and Detailed Syllabus of the MCA

Programme- specific Outcomes

- *Two years Post-Graduate programme have been designed as a base for research and application of the concepts of Data Science. This two years programme will enable the students to enhance their learning after under-graduate course and to join the workforce in the field of Data Science.*
- *Two years Post-Graduate programme will enable the students with the understanding of the Development of sustainable cities and communities by the concept of green computing. After two years, students will be able to initiate industry-academia collaboration for innovative research to maintain the sustainability.*
- *Two years Post-Graduate programmes in Geoinformatics will enable the students with recent development in the field of Machine learning and Internet-of-Things (IoT). After two years, students will be able to apply these concepts and implements several devices that can help in regular operations and alarming situations in several industries.*

Global needs	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201
System thinking competency	√			√		√		√		√			√		√	
Anticipatory Competency		√			√			√		√		√				√
Normative Competency	√			√		√		√		√		√			√	√
Strategic Competency	√				√		√				√			√		
Transdisciplinary Collaboration Competency	√		√		√			√	√				√		√	
Critical Thinking Competency		√		√		√			√	√		√		√		√

Creativity Competency			√		√		√		√					√	
Self-Awareness Competency	√			√			√				√		√		√
Integrated Problem-solving Competency		√		√	√	√		√	√			√		√	√

Global needs	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	
System thinking competency	√		√		√			√		√		√			√			√			√
Anticipatory Competency		√		√		√		√		√		√			√	√				√	
Normative Competency	√			√		√		√		√		√			√		√				
Strategic Competency	√		√		√		√		√		√		√			√				√	
Transdisciplinary Collaboration Competency		√			√	√		√	√	√		√		√	√		√			√	
Critical Thinking Competency	√			√		√		√		√		√		√	√			√			√
Creativity Competency		√	√			√		√			√	√			√			√			
Self-Awareness Competency	√	√		√			√		√			√		√			√		√		√
Integrated Problem-solving Competency		√	√		√	√	√		√	√			√	√		√		√	√		√

National needs	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT101	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	MCAMT201	
Promote Right to education	√				√				√		√		√					√			
Inculcate ethical and professional values			√				√				√	√	√								

Increase national and international visibility;		√			√					√	√			√		√
Leverage institutional strengths through strategic partnerships;	√			√		√		√			√		√			
Enlarge the academic community within which to benchmark their activities;	√	√	√	√		√		√			√	√		√		
Mobilise internal intellectual resources;					√			√		√			√	√		
Add important, contemporary learning outcomes to student experience;	√	√	√		√	√		√	√		√	√	√	√		√
Develop stronger research groups.	√		√	√				√		√					√	
Encourage multidisciplinary			√		√			√		√	√		√	√		
Promote Cross cultural exchanges										√			√			
Preservation of traditional knowledge		√	√		√			√	√			√	√		√	√
Creating human resource for Economic growth	√			√		√		√		√			√	√	√	
Promotion of scientific mind-set and critical thinking	√		√		√	√	√	√	√	√	√					√

National needs	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	
Promote Right to education					√	√	√	√		√	√		√		√							√
Inculcate ethical and professional values	√		√		√		√		√	√	√		√	√			√		√		√	
Increase national and international visibility	√		√	√	√		√	√	√		√		√	√	√		√		√			√
Leverage institutional strengths through strategic partnerships			√	√	√		√	√	√			√		√	√		√			√		√
Enlarge the academic community within which to benchmark their activities	√		√		√		√	√			√	√		√	√				√			
Mobilise internal intellectual	√			√		√		√	√		√	√	√		√				√			

and communities

Sustainable development needs:	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAMT301	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	MCAIT401	
Help to eradicate poverty	√			√					√	√				√					√			√
Ensuring meal for all	√	√				√			√	√			√		√							
Promoting good health and well being	√			√						√				√					√			
Promoting quality education	√		√	√		√	√			√	√	√	√	√	√				√			√
Promoting gender equality	√				√	√		√		√			√		√				√			√
Initiatives for clean water and sanitization		√	√			√	√	√	√				√	√					√			√
Programmes to reduce inequalities	√					√			√				√	√					√			
Develop sustainable cities and communities		√	√			√	√	√	√		√	√				√	√			√		

Curriculum of MCA



Course of Study Code	Discipline Code	Discipline	Course Name	Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	Internal Marks	End Sem Marks	Total Marks	Sem Marks			
MCA	MCA	Master of Computer Application	I	Numerical Methods	MT	MCAMT101	MT-1	2-1-0	3	22	15	35	50	400		
				Financial & Organizational Management	BM	MCABM101	BM-1	2-1-0	3		15	35	50			
				Digital Electronics & Computer Organization	IT	MCAIT101	IT-1	2-1-0	3		15	35	50			
				Database Management System	IT	MCAIT102	IT-2	2-1-0	3		15	35	50			
				Design & Analysis of Algorithms	IT	MCAIT103	IT-3	2-1-0	3		15	35	50			
				Linux & Shell Programming	IT	MCAIT104	IT-4	2-1-0	3		15	35	50			
				Lab in JavaScript, HTML and CSS	ITL	MCAITL101	ITL-1	0-0-4	2		30	20	50			
				Lab in Framework and Linux	ITL	MCAITL102	ITL-2	0-0-4	2		30	20	50			
			II	Theory of Computation	MT	MCAMT201	MT-2	2-1-0	3	22	15	35	50	400		
				Systems Programming	IT	MCAIT201	IT-5	2-1-0	3		15	35	50			
				Core & Advanced JAVA	IT	MCAIT202	IT-6	2-1-0	3		15	35	50			
				Computer Graphics	IT	MCAIT203	IT-7	2-1-0	3		15	35	50			
				Software Engineering & Case Tools	IT	MCAIT204	IT-8	2-1-0	3		15	35	50			
				Computer Architecture	IT	MCAIT205	IT-9	2-1-0	3		15	35	50			
				Lab in Assembly Language Programming and Java	ITL	MCAITL201	ITL-3	0-0-4	2		30	20	50			
				Lab in Computer Graphics and Case Tools	ITL	MCAITL202	ITL-4	0-0-4	2		30	20	50			
			III	Operation Research	MT	MCAMT301	MT-3	2-1-0	3	22	15	35	50	400		
				.NET Framework & C#	IT	MCAIT301	IT-10	2-1-0	3		15	35	50			
				Data Communication & Networks	IT	MCAIT302	IT-11	2-1-0	3		15	35	50			
				Information Retrieval & Web Mining	IT	MCAIT303	IT-12	2-1-0	3		15	35	50			
				Machine Learning Techniques	IT	MCAIT304	IT-13	2-1-0	3		15	35	50			
				Multimedia Systems	IT	MCAIT305	IT-14	2-1-0	3		15	35	50			
				MAIN PROJECT (Phase I): Dissertation and Internal Assessment	ITP	MCAITP301	ITP-1	0-0-4	2		30	20	50			
				MAIN PROJECT (Phase I): Viva-voce	ITP	MCAITP302	ITP-2	0-0-4	2		30	20	50			
			IV	Secure Computing	IT	MCAIT401	IT-15	2-1-0	3	22	15	35	50	500		
				Embedded System Design	IT	MCAIT402	IT-16	2-1-0	3		15	35	50			
				Digital Communication	IT	MCAIT403	IT-17	2-1-0	3		15	35	50			
				Image Processing	IT	MCAIT404	IT-18	2-1-0	3		15	35	50			
				Data Science & Python/R	ITE (Any Two)	MCAITE401	ITE-1,ITE-2	2-1-0	3+3+6		15	35	50+50+ 100			
				Robotics & AI		MCAITE402		2-1-0			15	35				
				Distributed Computing		MCAITE403		2-1-0			15	35				
				Cloud Computing		MCAITE404		2-1-0			15	35				
				Mobile Computing & Applications		MCAITE405		2-1-0			15	35				
				Big Data Analysis		MCAITE406		2-1-0			15	35				
				Artificial Neural Networks		MCAITE407		2-1-0			15	35				
				Computer Vision		MCAITE408		2-1-0			15	35				
				High Performance Computing		MCAITE409		2-1-0			15	35				
				MAIN PROJECT (Phase II): MAIN PROJECT (Phase II): Viva-voce		ITP		MCAITP401			ITP-3	0-0-4			2	60
				MAIN PROJECT (Phase II): Viva-voce	ITP	MCAITP402	ITP-4	0-0-4	2		60	40	100			
				Total Credit / Marks									88			

Abbreviations: MT = Mathematics Course; BM = Business Management Course; IT = Information Technology Course; ITL = Information Technology Laboratory; ITP = Information Technology Project; ITE = Information Technology Elective Course; L= Lecture Hour; T= Tutorial Hour; P= Practical Hour



**DEPARTMENT OF COMPUTER SCIENCE
KAZI NAZRUL UNIVERSITY**

P.S, Nazrul Road, Kalla Bypass More, North,
P.O, C H Kalla, Asansol, West Bengal 713340

Syllabus of MCA
(Applicable from 2022-23 onwards)

Course Name: MCA

Course Structure:

Summary:

Semester No	Total Credit
1	22
2	22
3	22
4	22
Total	88

Detailed Course Structure

Semester – I

Course Name	Course Type	Course Code	L - T - P	Course Credit	Sem Credit
Numerical Methods	MT	MCAMT101	2 - 1 - 0	3	22
Financial & Organizational Management	BM	MCABM101	2 - 1 - 0	3	
Digital Electronics & Computer Organization	IT	MCAIT101	2 - 1 - 0	3	
Database Management System	IT	MCAIT102	2 - 1 - 0	3	
Design & Analysis of Algorithms	IT	MCAIT103	2 - 1 - 0	3	
Linux & Shell Programming	IT	MCAIT104	2 - 1 - 0	3	
Lab in JavaScript, HTML and CSS	ITL	MCAITL101	0 - 0 - 4	2	
Lab in Framework and Linux	ITL	MCAITL102	0 - 0 - 4	2	

Semester – II

Course Name	Course Type	Course Code	L - T - P	Course Credit	Sem Credit
Theory of Computation	MT	MCAMT201	2 - 1 - 0	3	22
Systems Programming	IT	MCAIT201	2 - 1 - 0	3	
Core & Advanced JAVA	IT	MCAIT202	2 - 1 - 0	3	
Computer Graphics	IT	MCAIT203	2 - 1 - 0	3	
Software Engineering & Case Tools	IT	MCAIT204	2 - 1 - 0	3	
Computer Architecture	IT	MCAIT205	2 - 1 - 0	3	
Lab in Assembly Language Programming and Java	ITL	MCAITL201	0 - 0 - 4	2	
Lab in Computer Graphics and Case Tools	ITL	MCAITL202	0 - 0 - 4	2	

Semester – III

Course Name	Course Type	Course Code	L - T - P	Course Credit	Sem Credit
Operation Research	MT	MCAMT301	2 - 1 - 0	3	22
.NET Framework & C#	IT	MCAIT301	2 - 1 - 0	3	
Data Communication & Networks	IT	MCAIT302	2 - 1 - 0	3	
Information Retrieval & Web Mining	IT	MCAIT303	2 - 1 - 0	3	
Machine Learning Techniques	IT	MCAIT304	2 - 1 - 0	3	
Multimedia Systems	IT	MCAIT305	2 - 1 - 0	3	
MAIN PROJECT (Phase I): Dissertation and Internal Assessment	ITP	MCAITP301	0 - 0 - 4	2	
MAIN PROJECT (Phase I): Viva-voce	ITP	MCAITP302	0 - 0 - 4	2	

Semester – IV

Course Name	Course Type	Course Code	L - T - P	Course Credit	Sem Credit
Secure Computing	IT	MCAIT401	2 - 1 - 0	3	22
Embedded System Design	IT	MCAIT402	2 - 1 - 0	3	
Digital Communication	IT	MCAIT403	2 - 1 - 0	3	
Image Processing	IT	MCAIT404	2 - 1 - 0	3	
Data Science & Python/R	ITE (Any Two)	MCAITE401	2 - 1 - 0	3+3=6	
Robotics & AI		MCAITE402	2 - 1 - 0		
Distributed Computing		MCAITE403	2 - 1 - 0		
Cloud Computing		MCAITE404	2 - 1 - 0		
Mobile Computing & Applications		MCAITE405	2 - 1 - 0		
Big Data Analysis		MCAITE406	2 - 1 - 0		
Artificial Neural Networks		MCAITE407	2 - 1 - 0		
Computer Vision		MCAITE408	2 - 1 - 0		
High Performance Computing		MCAITE409	2 - 1 - 0		
MAIN PROJECT (Phase II): Dissertation and Internal Assessment	ITP	MCAITP401	0 - 0 - 4	2	
MAIN PROJECT (Phase II): Viva-voce	ITP	MCAITP402	0 - 0 - 4	2	

Semester - I

Course Name: Numerical Methods

Course Code: MCAMT101

Course Type: Theory	Course Details: MT- 1		L-T-P: 2 - 1 - 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

1. Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
2. Apply numerical methods to obtain approximate solutions to mathematical problems.
3. Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
4. Analyse and evaluate the accuracy of common numerical methods.

Detailed contents:

1. **Floating point Arithmetic:** Numeric computation and error analysis; Representation of floating point numbers, Operations, Normalization, Pitfalls of floating point representation; Iterative Methods: Zeros of a single transcendental equation and zeros of polynomial using Bisection Method, Iteration Method, Regula-Falsi method, Newton Raphson method, Secant method, Rate of convergence of iterative methods. (8 Lectures)
2. **Simultaneous Linear Equations:** Solutions of system of Linear equations, Gauss Elimination direct method and pivoting, Ill Conditioned system of equations, Refinement of solution; Gauss Seidal iterative method, Rate of Convergence. (8 Lectures)
3. **Interpolation and approximation:** Finite Differences, Difference tables; Polynomial Interpolation: Newton's forward and backward formula; Central Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula; Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation Approximation of function by Taylor's series and Chebyshev polynomial. (8 Lectures)
4. **Numerical Differentiation and Integration:** Introduction, Numerical Differentiation, Numerical Integration, Trapezoidal rule, Simpson's rules, Boole's Rule, Weddle's Rule Euler- Maclaurin Formula; Solution of differential equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta methods, Predictor-corrector method, Automatic error monitoring, stability of solution. (8 Lectures)
5. **Curve fitting, Cubic Spline and Approximation:** Method of least squares, fitting of straight lines, polynomials, exponential curves etc; Frequency Chart: Different frequency chart like Histogram, Frequency curve, Pi-chart; Regression analysis: Linear and Non-linear regression, Multiple regression; Time series and forecasting: Moving averages, smoothing of curves, forecasting models and methods; Statistical Quality Controls methods. (8 Lectures)

References :

1. C.Xavier: C Language and Numerical Methods.
2. Dutta & Jana: Introductory Numerical Analysis.
3. J.B.Scarborough: Numerical Mathematical Analysis.
4. Jain, Iyengar , & Jain: Numerical Methods (Problems and Solution).
5. Balagurusamy: Numerical Methods, Scitech.
6. Baburam: Numerical Methods, Pearson Education.
7. N. Dutta: Computer Programming & Numerical Analysis, Universities Press.
8. Soumen Guha & Rajesh Srivastava: Numerical Methods, OUP.
9. Srimanta Pal: Numerical Methods, OUP.

Course Name: Financial & Organizational Management

Course Code: MCABM101

Course Type: Theory	Course Details: BM- 1		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

On successful completion of this module, the learner will be able to:

1. Describe the financial environment within which organisations must operate
2. Critically evaluate the financial objectives of various types of organisations and the respective requirements of stakeholders
3. Discuss the function of capital markets
4. Explain alternative sources of finance and investment opportunities and their suitability in particular circumstances
5. Assess the factors affecting investment decisions and opportunities presented to an organisation
6. Select and apply techniques in managing working capital
7. Analyse a company's performance and make appropriate recommendations.

Detailed contents:

1. **Introduction to Financial Accounting:** Principles, concepts and conventions, Double entry system, Ledger, Trial balance. Final accounts: Trading, profit and loss accounts and balance sheet, Basic Company related concepts; Shares, Debentures, Preference Shares. (8 Lectures)
2. **Introduction to Financial Management:** Basic Financial Management Concepts, Capital Budgeting; Meaning, Importance, Techniques, Ratio analysis: Meaning, Types of Ratios, Usefulness: Introduction to Computerized Accounting System: Data Collection process, Coding logic and codes required, Master files, Transaction files. (10 Lectures)
3. **Organization Structure and Management:** Theories of Management: Fayol's principles of Management, Organization and organization Structure: Line and Staff authority, Centralization and Decentralization, Span of control: Types of organization structure; function based, Product based, Geography based, Project based: Organization Design: Mechanistic and Organic, Virtual and Network. (10 Lectures)
4. **Management Information Systems:** Definition: Types of information system, MIS versus Data processing, MIS & Decision Support Systems, MIS & Information Resources Management, Structure of Management information system. Concepts of planning & control in organizational: The Planning Process, Computational support for planning control process. (6 Lectures)
5. **Advanced Concepts in Information Systems:** Enterprise Resource Planning, Supply Chain Management, Customer Relationship Management and Procurement Management. Security and Ethical challenges. (6 Lectures)

References:

1. S.N. Maheswari & S. K. Maheswari, "Introduction to Financial Accountancy", Vikas Publication.
2. S.N. Maheswari & S. K. Maheswari, "Advanced Accountancy", Vikas Publication.
3. S.N. Maheswari & S. K. Maheswari, "Financial Management", Viaks Publication.
4. O Brian, "Introduction to Information System", MCGRAW HILL.
5. Murdick, "Information System for Modern Management", PHI.
6. Jawadekar, " Management Information System", TMH.
7. Jain Sarika, "Information System", PPM
8. Davis, "Information System", Palgrave Macmillan
9. L. M. Prasad, "Organizational Behavior", S. Chand
10. Koontz, Weihrich, "Essentials of Management", TMH
11. K. Aswathappa, "Human Resource and Personnel Management", TMH
12. L. M. Prasad, "Human Resource Management", S. Chand

Course Name: Digital Electronics & Computer Organization

Course Code: MCAIT101

Course Type: Theory	Course Details: IT- 1		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

1. Recall and summarize the basic concept of computer fundamentals, Number system, Boolean algebra, Karnaugh map and Perform problems on IEEE 754 standard number system
2. Explain the concept of stored program, role of operating system, Instruction sets and Addressing modes and Demonstrate problems on Addressing modes

3. Design of adders, ALU and Memory management unit and illustrate problems related to cache memory
4. Explain and Use fixed point multiplication (Booth's) and division (Restoring and non-restoring) algorithms
5. Explain the concept of Instruction pipeline, RISC, CISC

Detailed contents:

1. **Representation of Information and Basic Building Blocks:** Introduction to Computer- Basic building blocks, Computer hardware and Software; Types of Computers; Number System: Binary, Octal, Hexadecimal, Gray code; Character Codes (BCD, ASCII, EBCDIC, UNICODE), Logic gates, Boolean Algebra; Logic expression –SOP & POS form; K-map simplification; Design of Combinatorial circuits: Half Adder, Full Adder, Subtractor, Decoder, Encoders, Multiplexer, Demultiplexer, Carry lookahead adder, Parity bit Generator & Checker; Design of Sequential circuits: Latches; Flip-Flops; Registers; Counters (synchronous & asynchronous); State reduction;ALU, Micro-Operations. (8 Lectures)
2. **Basic Organization:** Stored Program concept (Von Neumann Machine); Operational flow chart (Fetch, Execute), Instruction Cycle, Organization of Central Processing Unit, Hardwired & micro programmed control unit, General Register Organization, Stack Organization, Addressing modes, Instruction formats, data transfer & Manipulation, I/O Organization, Bus Architecture, Programming Registers. ALU- chip, Faster Algorithm and Implementation (multiplication & Division) (8 Lectures)
3. **Memory Organization:** Memory Hierarchy, Main memory (RAM/ROM chips), Types of memory; Auxiliary memory - magnetic disk and its performance, magnetic Tape, Optical memory; Associative memory, Cache memory, Virtual Memory, Memory Management Hardware, hit/miss ratio; Memory interleaving. (8 Lectures)
4. **I/O Organization:** Peripheral devices, I/O interface methods, Interrupts and its usage; Modes of Transfer, Priority Interrupt, Direct Memory Access, Input-Output Processor, and Serial Communication. I/O Controllers, Asynchronous data transfer, Strobe Control, Handshaking. (8 Lectures)
5. **Process Organization:** Basic functional block diagram of a computer; Introduction to microprocessor – Essential & Non-essential components; Basic Concept of 8-bit micro Processor (8085) and 16-bit Micro Processor (8086), Instruction Set, Assembly language program of (8085): Addition of two numbers, Subtraction, Block Transfer, Finding greatest number, Table search, Numeric Manipulation; Comparison with other 8/16 bit microprocessors; Introduction to microcontroller; Coprocessors; Introductory Concept of pipeline. (8 Lectures)

References:

1. William Stalling, "Computer Organization & Architecture", Pearson education Asia
2. Mano Morris, "Computer System Architecture", PHI
3. Zaky & Hamacher, "Computer Organization", McGraw Hill
4. Ram, "Computer Fundamental Architecture & Organization", New Age
5. Tannenbaum, "Structured Computer Organization", PHI.

Course Name: Database Management System

Course Code: MCAIT102

Course Type: Theory	Course Details: IT- 2		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

1. Study the physical and logical database designs, database modeling, relational, hierarchical, and network models
2. Understand the different issues involved in the design and implementation of a database system.
3. Understand and use data manipulation language to query, update, and manage a database

4. Develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency,
5. Design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Detailed contents:

- 1. Introduction and ER Modelling:** An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure. Data Modelling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree. (8 Lectures)
- 2. Relational data Model and Language:** Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus. (8 Lectures)
- 3. Introduction to SQL:** Characteristics of SQL, Advantages of SQL, SQL data types and literals, Types of SQL commands, SQL operators and their procedure, Tables, views and indexes, Queries and sub queries, Aggregate functions, Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors in SQL. PL/SQL, Triggers and clusters. (8 Lectures)
- 4. Data Base Design & Normalization:** Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependencies, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design. (8 Lectures)
- 5. Transaction Processing:** Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling. Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi-version schemes, Recovery with concurrent transaction; Transaction processing in Distributed system, data fragmentation. Replication and allocation techniques for distributed system, overview of concurrency control and recovery in distrusted database. (8 Lectures)

References:

- 1 Date C J, "An Introduction To Database System", Addison Wesley
- 2 Korth, Silbertz, Sudarshan, "Database Concepts", McGraw Hill
- 3 Elmasri, Navathe, "Fundamentals Of Database Systems", Addison Wesley
- 4 Paul Beynon Davies, "Database Systems", Palgrave Macmillan

Course Name: Design & Analysis of Algorithms

Course Code: MCAIT103

Course Type: Theory	Course Details: IT- 3		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be

1. Able to Argue the correctness of algorithms using inductive proofs and Analyze worst-case running times of algorithms using asymptotic analysis.
2. Able to explain important algorithmic design paradigms (divide-and-conquer, greedy method, dynamic-programming and Backtracking) and apply when an algorithmic design situation calls for it.
3. Able to Explain the major graph algorithms and Employ graphs to model engineering problems, when appropriate.
4. Able to Compare between different data structures and pick an appropriate data structure for a design situation.
5. Able to Describe the classes P, NP, and NP Complete and be able to prove that a certain problem is NP-Complete.

6. Able to analyze String matching algorithms.

Detailed contents:

1. **Introduction:** Algorithms, Analysis of Algorithms, Design of Algorithms, and Complexity of Algorithms, Asymptotic Notations, Growth of function, Recurrences Sorting in polynomial Time: Insertion sort, Merge sort, Heap sort, and Quick sort Sorting in Linear Time: Counting sort, Radix Sort, Bucket Sort. (8 Lectures)
2. **Divide and Conquer:** Binary search, Strassen’s matrix multiplication, Greedy Algorithm: General Approach, Knapsack Problem, Huffman Code; Number Theoretic Algorithms: Factorization, GCD, Modular Arithmetic; StringMatching: Naive, Rabin-Karp, KMP Algorithms. (8 Lectures)
3. **Advanced Design and Analysis Techniques: Dynamic programming:** General approach, Multistage graph, Matrix- chain multiplication, Longest common subsequence, 0/1 Knapsack problem; **Backtracking:** N- queen problem, Sum of subsets, Knapsack problem, Travelling salesperson problem; **Branch-and-Bound:** Assignment problem, 0/1 knapsack problem; Amortized Analysis. (8 Lectures)
4. **Graph Algorithms:** Breadth first search, depth first search, **Application of BFS, DFS:** Connected component, Topological sort, **Minimum Spanning Tree:** Kruskal's Algorithms, Prim's Algorithms, **Shortest Path: Single Source:** Dijkstra, Bellman Ford; **All pair shortest Path:** Floyd's Algorithms, **Maximum flow:** Flow networks and Ford Fulkerson algorithms (8 Lectures)
5. **Theory of NP-Completeness:** P, NP, NP-Complete, NP Hard, **Randomized Algorithms:** Randomized Quick- Sort, Randomized algorithms for n-queens, randomized min cut; **Approximation Algorithms:** Travelling Salesman Problem, Bin packing, Set cover, max cut; **Computational Geometry:** Convex Hull; **Parallel Algorithms.** (8 Lectures)

References:

1. Horowitz Sahani, “ Fundamentals of Computer Algorithms”, Goltotia
2. Cormen Leiserson etal, “ Introduction to Algorithms”, PHI
3. Brassard Bratley, “Fundamental of Algorithms”, PHI
4. M T Goodrich etal, “Algorithms Design”, John Wiley
5. A V Aho etal, “The Design and analysis of Algorithms”, Pearson Education

Course Name: Linux & Shell Programming

Course Code: MCAIT104

Course Type: Theory	Course Details: IT- 4		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to

1. Discuss the history and development of the Linux Operating System.
2. Work within a command line environment
3. Describe and apply various command line utilities
4. Work with the file System
5. Work with shells
6. Write shell scripts
7. Perform System Administration Tasks

Detailed contents:

1. **Introduction:** Introduction to LINUX, LINUX system organization (the kernel and the shell), Files and directories,

- Library functions and system calls, Editors (vi and ed). (8 Lectures)
- LINUX Shell programming:** Types of Shells, Shell Meta characters, Shell variables, Shell scripts, Shell commands, the environment, Integer arithmetic and string Manipulation, Special command line characters, Decision making and Loop control, controlling terminal input, trapping signals, arrays. (8 Lectures)
 - Portability with C:** Command line Argument, Background processes, process synchronization, Sharing of data,user-id, group-id, pipes, fifos, message queues. (8 Lectures)
 - LINUX System Administration:** File System, mounting and unmounting file system, System booting, shutting down, handling user account, backup, recovery, security, creating files, storage of Files, Disk related commands. (8 Lectures)
 - Different tools and Debugger:** System development tools: lint, make, SCCS (source code control system), Language development tools: YACC, LEX, M4, Text formatting tools: nroff, troff, tbl, eqn, pic, Debugger tools: Dbx and Adb. (8 Lectures)

References:

- Linux Networking & System Administration, Terry Collings and Kurt Wall (Wiley)
- Red Hat Linux 9, Bill Ball and Hoyt Duff (Pearson Education)
- Unix Shell Programming, Yashavant Kanetkar (BPB Publications)
- Unix & Shell Programming, Sumitabha Das, (Tata McGraw Hill)

Course Name: Lab in JavaScript, HTML and CSS

Course Code: MCAITL101

Course Type: Practical	Course Details: ITL- 1		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		30	20

Course Learning Outcomes:

Students will be able to

- Perform basic and data format form validation.
- Handle errors and exceptions.
- Create canvas and draw graphics using Javascript.
- Create complex Animations using CSS and Javascript.
- Create Image Maps using Javascript.
- Build a 2D game using CSS and Javascript.

Course Name: Lab in Framework and Linux

Course Code: MCAITL102

Course Type: Practical	Course Details: ITL- 2		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		30	20

Course Learning Outcomes:

Students will be able to

- Work within a command line environment

2. Work with the file System
3. Work with shells
4. Write shell scripts
5. Perform System Administration Tasks

Semester – II

Course Name: Theory of Computation

Course Code: MCAMT201

Course Type: Theory	Course Details: MT- 2		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

At successful completion of the course, students should:

1. Demonstrate advanced knowledge of formal computation and its relationship to languages
2. Distinguish different computing languages and classify their respective types
3. Recognise and comprehend formal reasoning about languages
4. Show a competent understanding of the basic concepts of complexity theory

Detailed contents:

- 1. Recursive Functions & Formal Languages:** Partial & total functions; Recursive functions; Bounded minimization; Ackermann's function; Strings; Free semi-group; Languages; Generative grammars and their languages; Chomsky classification of grammars and languages. (8 Lectures)
- 2. Finite Automata:** Deterministic & Non-deterministic finite automata; Machines with moves on empty strings, regular expressions, Regular sets, Relationship with regular grammars, Pumping lemma for regular sets and its usage, Closure property of regular sets, Decision algorithm for regular sets, Minimization of Finite Automata. (8 Lectures)
- 3. Context Free Grammars:** Derivation Trees, Simplification of context free grammars, Chomsky normal form, Greibach normal form, Decision algorithm. (8 Lectures)
- 4. Pushdown Automata:** Instantaneous description, languages accepted by final states and empty stacks, deterministic pushdown automata, relationship with context free language. (8 Lectures)
- 5. Turing Machines and Undecidability:** Instantaneous description, Languages, String manipulation, Turing computability of functions, Equivalence between Turing Computability and partial recursiveness. **Undecidability:** Recursively enumerable and recursively decidable languages, Undecidability of decision algorithm for Type 0 grammar, Church Turing thesis, Halting problem of machine. (8 Lectures)

References:

1. Automata, Languages & Computation - Hopcraft & Ullman
2. Theory of computability - Hennie
3. Formal languages - Revesz
4. Theory of Computer Science - K L P Mishra & Chandrashekharan

Course Name: Systems Programming

Course Code: MCAIT201

Course Type: Theory	Course Details: IT- 5		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

At successful completion of the course, students should have

1. Adequate knowledge in system programs (assemblers, loaders, linkers, macro-processors, text editors, debuggers, interpreters, compilers, operating systems).
2. Ability to use theoretical and applied information in these areas to design system software with realistic constraints.
3. Ability to conduct experiments, gather data, analyze and interpret results for investigating solutions to real life applications with assembly language programming and Unix shell programming.
4. Ability to devise, select, and use modern techniques and tools needed for the design and implementation of system programs.
5. Ability to work efficiently in intra-disciplinary teams and to work individually.

Detailed contents:

1. **Language Processors:** Introduction; Language Processing Activities; Fundamentals of Language Processing; Fundamentals of Language Specifications; Data Structures for Language Processing; Scanning & Parsing. (6 Lectures)
2. **Assemblers:** Elements of Assembly Language Programming; A Simple Assembly Scheme; Pass Structure of Assemblers; Types of Assemblers; Assembler Directives; Design of a Two Pass Assembler; Macros and Macro Processors; Nested Macro Calls; Advanced Macro Facilities; Design of a Macro Pre-processor. (6 Lectures)
3. **Lexical and Syntax Analysis:** Aspects of Compilation; Memory Allocation; Lexical analysis – DFA diagrams; Syntactic analysis- Bottom up & Top-Down Parsing; Symbol Tables – Hash & Tree Structures, Parse trees. (6 Lectures)
4. **Semantic analysis and Code Optimization:** Syntax Directed Translation; Compilation of Expressions; Compilation of Control Structures; Code generation & Optimization; Register utilization. (6 Lectures)
5. **Linkers:** Relocation & Linking concepts; Design of a Linker; Self relocating Programs; A Linker for MS-DOS; Linking for overlays; Loaders; Types of Loaders & Linkers. (6 Lectures)
6. **Introduction to Operating Systems:** Definition and types of operating systems; Definition & modules of OS; Process concept, Process scheduling, Threads, Inter-process communication, CPU scheduling; Scheduling algorithms, Multiple-processor scheduling; Real-time scheduling; Process synchronization & Deadlocks; Memory management techniques; I/O Management techniques; Security issues. (10 Lectures)

References:

1. Abraham Siberschatz and Peter Baer Galvin, "Operating System Concepts", Fifth Edition, Addison-Wesley
2. Milan Milankovic, "Operating Systems, Concepts and Design", McGraw-Hill.
3. Harvey M Deital, "Operating Systems", Addison Wesley
4. Richard Peterson, "Linux: The Complete Reference", Osborne McGraw-Hill.
5. Hopcroft, Sethi and Ullman, Compiler Principles, AddisonWesley
6. John Levine, Linkers and Loaders, <http://www.iecc.com>
7. H. Abelson and G. Sussmann, Structure and Interpretation of Computer Programs, MIT Press

8. Hopcroft and Ullman, Introduction to Automata theory, Languages and Computation, Narosa Pub.
9. Systems Programming & Operating Systems- D M Dhamdhere THM

Course Name: Core & Advanced JAVA

Course Code: MCAIT202

Course Type: Theory	Course Details: IT- 6		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. Summarize the strengths and weaknesses of Java programming and the basic concepts of object-oriented programming.
2. Identify Java code utilities in applets, Java packages, and classes.
3. Write Java code using advanced Java features.

Detailed contents:

1. **Introduction to object-oriented programming:** Features of Java; constants, variables and data types, Operators and expressions; decision making, branching and looping; Classes, objects and Methods; arrays, strings and vectors; Java Applet, String handling, Networking, Event handling, Introduction to AWT, AWT controls, Layout managers, Menus, Images, Graphics. (6 Lectures)
2. **Inheritance & File Handling:** Interfaces; Managing Input/ Output Files in Java; Packages, Exception Handling, Multithreaded Programming; Utility Classes; String Handling; Generics, Generic Class, Generic methods. (6 Lectures)
3. **JDBC Overview:** JDBC implementation; Connection class; Statements; Catching Database Results, handling database Queries. Networking; Inet Address class; URL class; TCP sockets; UDP sockets. (6 Lectures)
4. **Web Page Design:** Web page Designing using HTML, Scripting basics: Client side and server-side scripting; Java Script- Object, names, literals, operators and expressions, statements and features, events, windows- documents, frames, data types, built-in functions, Browser object model, verifying forms; Servlet: life cycle of a servlet; The Servlet API, Handling HTTP Request and Response, using Cookies, Session Tracking; Introduction to JSP. (6 Lectures)
5. **XML:** Introduction, Displaying an XML Document, Data Interchange with an XML document, Document type definitions, Parsers using XML, Client-side usage, Server-Side usage; Common Gateway Interface (CGI), PERL, RMI, COM/DCOM, VBScript, Active Server Pages (ASP). (6 Lectures)
6. **Advanced Java:** Steps to create JDBC Application- Import the packages, Register the JDBC driver, Open a connection, Execute a query, Extract data from result set, Clean up the environment. JDBC Driver Types: Type1: JDBC-ODBC Bridge Driver, Type 2: JDBC-Native API, Type 3: JDBC-Net pure Java, Type 4: 100% Pure Java. JDBC Connections: Import JDBC Packages, Register JDBC Driver; Introduction to Hibernate: Advantages of Hibernate compared to JDBC, Hibernate architecture, Hibernate Data Types, Hibernate API, Introduction to RMI; Introduction to Servlets: Servlet Life Cycle, Steps to create Servlet, Session Tracking; Introduction to Java Server Pages: Advantages of JSP, Life Cycle of JSP, JSP Scripting Elements, JSP Implicit Objects, JSP directives- page directive, include directive, taglib directive, Exception Handling in JSP, JSP Action Tags, MVCin JSP, JSTL (JSP Standard Tag Library): Advantage of JSTL, JSTL Tags (10 Lectures)

References:

1. Burdman, "Collaborative Web Development", Addison Wesley.
2. Sharma & Sharma, "Developing E-Commerce Sites", Addison Wesley
3. Ivan Bayross, "Web Technologies Part II", BPB Publications.
4. Margaret Levine Young, "The Complete Reference Internet", TMH
5. Naughton, Schildt, "The Complete Reference JAVA2", TMH
6. Balagurusamy E, "Programming in JAVA", TMH
7. Shishir Gundavarma, "CGI Programming on the World Wide Web", O'Reilly & Associate.
8. DON Box, "Essential COM", Addison Wesley.
9. Greg Buczek, "ASP Developer's Guide", TMH

Course Name: Computer Graphics

Course Code: MCAIT203

Course Type: Theory	Course Details: IT- 7		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. Describe the fundamental algorithms used in computer graphics and to some extent be able to compare and evaluate them
2. Work and interact, through hands-on experiences, to design, develop, and modify electronically generated imaginary using a wide range of sophisticated graphical tools and techniques
3. Summarize different hidden surface elimination algorithms and shading techniques used in computer graphics and digital media production.
4. Explain about the technology necessary for creating multimedia content for the web, video, DVD, 2D and 3D graphics, Sound and programming.
5. Apply the knowledge, techniques, skills and modern tools to become successful professionals in communication and media industries

Detailed contents:

1. **Introduction:** Application areas of Computer Graphics, overview of graphics systems, Video -display devices, Raster - scan systems, random scan systems, graphics monitors and work stations and input devices Output primitives: Points and lines, line drawing algorithms, mid - point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary - fill and flood - fill algorithms. (8 Lectures)
2. **2-D Geometrical transforms:** Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms transformations between coordinate systems.

2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to view - port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland -Hodgeman polygon clipping algorithm. (8 Lectures)
3. **Plane Curves and Surfaces:** Parametric and Non-parametric curves and their representations; Cubic Splines; Bezier and B Splines. B Spline and Bezier Surfaces; Visible line and surfaces detection algorithms: ZBuffer Algorithm; A Buffer Algorithm; Rendering; Phong reflection model, Incremental shading techniques; Gourard and Phong Shading; Illumination models; Shadows and Texture. (8 Lectures)
4. **3-D Geometric transformations:** Translation, rotation, scaling, reflection and shear transformations, composite transformations. 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping. (8 Lectures)
5. **Computer animation:** Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

Visible surface detection methods: Classification, back - face detection, depth - buffer, scan - line, depth sorting, BSP - tree methods, area sub- division and octree methods Illumination Models and Surface rendering Methods: Basic

illumination models, polygon rendering methods (8 Lectures)

References:

1. Computer Graphics: Principles and Practice: Foley et al.
2. Computer Graphics: Hern and Baker
3. Procedural elements in Computer Graphics: David F. Rogers
4. Mathematical Elements for Computer Graphics: David F. Rogers and J.A.Adams
5. Graph Theory with applications to engineering & computer science: Deo Narsing
6. Combinatorics and Graph Theory: Harris, John, Hirst, Jeffrey L., Mossinghoff, Michael

Course Name: Software Engineering & Case Tools

Course Code: MCAIT204

Course Type: Theory	Course Details: IT- 8		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Student will be able to

1. Acquire strong fundamental knowledge in science, mathematics, fundamentals of computer science, software engineering and multidisciplinary engineering to begin in practice as a software engineer.
2. Design applicable solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal and economic concerns.
3. Deliver quality software products by possessing the leadership skills as an individual or contributing to the team development and demonstrating effective and modern working strategies by applying both communication and negotiation management skill.
4. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development.

Detailed contents:

- 1. Introduction and SRS:** Introduction to software engineering, Importance of software, The evolving role of software, Software Characteristics, Software Components, Software Applications, Software Crisis, Software engineering problems, Software Development Life Cycle, Software Process. Requirement Analysis and Specification: Analysis Principles, Water Fall Model, The Incremental Model, Prototyping, Spiral Model, Role of management in software development, Role of matrices and Measurement, Problem Analysis, Requirement specification, Monitoring and Control. (8 Lectures)
- 2. Software-Design:** Design principles, problem partitioning, abstraction, top down and bottom up-design, Structured approach, functional versus object oriented approach, design specifications and verification, Monitoring and control, Cohesiveness, coupling, Forth generation techniques, Functional independence, Software Architecture, Transaction and Transform Mapping, Component – level Design, Forth Generation Techniques. (8 Lectures)
- 3. Coding and Testing:** Top-Down and Bottom –Up programming, structured programming, information hiding, programming style and internal documentation. **Testing:** Testing principles, Levels of testing, functional testing, structural testing, test plane, test case specification, reliability assessment, software testing strategies, Verification & validation, Unit testing, Integration Testing, Alpha & Beta testing, system testing and debugging. (8 Lectures)
- 4. Software Project Management:** The Management spectrum- (The people, the product, the process, the project), cost estimation, project scheduling, staffing, software configuration management, Structured Vs. Unstructured maintenance, quality assurance, project monitoring, risk management. (8 Lectures)
- 5. Software Reliability & Quality Assurance:** Reliability issues, Reliability metrics, Reliability growth modeling, Software quality, ISO 9000 certification for software industry, SEI capability maturity model, comparison between ISO & SEI CMM. **CASE (Computer Aided Software Engineering):** CASE and its Scope, CASE support in software life cycle, documentation, project management, internal interface, Reverse Software Engineering, Architecture of CASE environment. (8 Lectures)

References:

1. Pressman, Roger S., "Software Engineering: A Practitioner's Approach Ed. Boston: McGraw Hill, 2001
2. Jalote, Pankaj, "Software Engineering Ed.2", New Delhi: Narosa 2002
3. Schaum's Series, "Software Engineering", TMH
4. Ghezzi, Carlo and Others, "Fundamentals of Software Engineering", PHI
5. Alexis, Leon and Mathews Leon, "Fundamental of Software Engineering", Vikas
6. Sommerville, Ian, "Software Engineering", AWL, 2000
7. Fairly, "Software Engineering", New Delhi: TMH
8. Pfleerger, S, "Software Engineering", Macmillan, 1987

Course Name: Computer Architecture

Course Code: MCAIT205

Course Type: Theory	Course Details: IT- 9		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Student will be able to understand

1. How a modern computer works
2. How it supports execution of software
3. How design alternatives affect performance and cost

Detailed contents:

- 1. Introduction to computer architecture:** Difference between scalar, scalar pipeline, scalar super-pipeline and superscalar architecture; Pipeline Processing: An overlapped parallelism, Instruction and Arithmetic pipelines; Pipeline hazards; Comparative study of 16 bit and 32-bit processor families; Basics of Pentium Processors; Comparative study of Microcontrollers; Future Trends. (8 Lectures)
- 2. Introduction to Parallel computing:** Parallelism in Uni-processor Systems, Parallel computer Structures, Architectural Classification schemes, Parallel processing applications. Case studies of vector processors, Array processors, Cube, Hypercube, Parallel algorithms on hyper cubes, Multiprocessor system, Multiprocessor interfacing schemes, Tightly & loosely coupled systems. (8 Lectures)
- 3. Principles of design:** Designing pipelined processors, Internal forwarding and register tagging, Hazard detection and resolution, Job sequencing and collision prevention, Concept of Array and Vector processing, Masking and Data routing: Concurrency in Pipelining and Vector Processing; Cache Coherence (8 Lectures)
- 4. Other architectures:** Concept of RISC systems; Data flow architectures; Comparison with control flowsystems Shuffle exchange and Omega Network, VLIW Architecture; EPIC Architecture. (8 Lectures)
- 5. Introduction to Parallel Algorithms:** Addition on Tree, Cube, Mesh, Linear Array, PSN, etc. Matrixmultiplication on Mesh, Cube, Torus, etc.; Parallel Sorting; Associative Processing. (8 Lectures)

References:

1. Computer Architecture & Parallel processing – Hwang & Briggs
2. Computer Architecture – Jean Loop Bear
3. Introduction to Distributed and Parallel computing- Crichlow
4. Designing Efficient Algorithms for parallel Computers- M.J.Quinn
5. Introduction to Parallel Algorithms- Joseph JA
6. The Design and Analysis of Parallel Algorithms- S.G.Akl
7. Computer Architecture & Organization – B Govindrajalu TMH

Course Name: Lab in Assembly Language Programming and Java

Course Code: MCAITL201

Course Type: Practical	Course Details: ITL- 3		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		30	20

Course Learning Outcomes:

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

1. understand basic assembly language syntax;
2. development of general programming skills
3. ability to run assembly language code
4. Write Java code using advanced Java features.

Course Name: Lab in Computer Graphics and Case Tools

Course Code: MCAITL202

Course Type: Practical	Course Details: ITL- 4		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		30	20

Course Learning Outcomes:

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

1. Understand how to generate line, circle and ellipse also how to create 2D object and various transformation techniques.
2. Understand various 3D Transformation techniques using OpenGL.
3. Understand multimedia compression techniques and applications.
4. apply the concept of modeling and mechanism involved in UML
5. Prepare case studies for analyzing modeling techniques

LIST OF EXPERIMENTS

Implement The Exercises From 1 To 4 Using C

1. Implementation of Algorithms for drawing 2D Primitives – Line (DDA, Bresenham) – and Circle (Midpoint)
2. Implementation of 2D Geometric transformations: Translation, Rotation, Scaling, Reflection, Shear.
3. Implementation of Composite 2D Transformations.
4. Implementation of Liang - Barsky Line Clipping.

Implement the exercises from 5 to 7 using OpenGL

5. Implementation of 3D Transformations - Translation, Rotation, Scaling

6. Implementation of 3D Projections – Parallel, Perspective
7. Creating 3D Scenes
8. Compression Algorithms - To implement text and image compression algorithms.
9. Image Editing and Manipulation - Basic Operations on image using any image editing software, Creating gif animated images, Image optimization
10. 2D Animation – To create Interactive animation using any authoring tool

Study of UML.

11. Create a UML model for Online Purchase system
12. Create a UML model for Library Management system.
13. Create a UML model for E-TICKETING.
14. Create a UML model for Quiz system.
15. Create a UML model for Student Mark Analyzing system
16. Create a UML model for E-Mail Client system
17. Create a UML model for Telephone phone Dialing.
18. Create a UML model for Point of Sale.
19. Create a UML model for Working Company.

Semester – III

Course Name: Operation Research

Course Code: MCAMT301

Course Type: Theory	Course Details: MT- 3		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

1. Identify and develop operational research models from the verbal description of the real system.
2. Understand the mathematical tools that are needed to solve optimisation problems.
3. Use mathematical software to solve the proposed models.
4. Develop a report that describes the model and the solving technique, analyse the results and propose recommendations in language understandable to the decision-making processes in Management and Engineering.

Detailed contents:

- 1. Network Analysis:** Terminology of network, shortest route problem, minimal spanning tree problem, max-flow problem. (8 Lectures)
- 2. PERT & CPM:** Project Scheduling by PERT, CPM: Diagram, representation, Critical path calculation, Construction of time chart and resource labelling, Probability and cost consideration in project scheduling, project control. (8 Lectures)
- 3. Linear and Nonlinear Programming:** Simplex Method, Revised simplex method, Duality in Linear programming, Application of Linear Programming to Economic and Industrial Problems. Nonlinear Programming: The Kuhn-Tucker conditions, Quadratic programming, convex programming. (8 Lectures)
- 4. Replacement Models:** Introduction, Replacement policies for items whose efficiency deteriorates with time, Replacement policies for items that fail completely. (8 Lectures)
- 5. Sequencing Model:** Classification of self- problems, processing of n jobs through two machines, threemachines, processing of two jobs through m machines. (8 Lectures)

References:

1. Taha, Operations Research, Macmillan.

2. B.E. Gillet, Introduction to Operations Research, McGraw-Hill.
3. S.S.Rao, Optimization Theory and Applications, Wiley Eastern.
4. G.Hadley, Linear programming, Addison-Wesley.

Course Name: .NET Framework & C#

Course Code: MCAIT301

Course Type: Theory	Course Details: IT- 10		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

After completion of the course the student will be able to

1. Understand the .NET framework.
2. Develop a proficiency in the C# programming language
3. Proficiently develop ASP.NET web applications using C#.
4. Use ADO.NET for data persistence in a web application.
5. Understand the 3-tier software architecture (presentation/client tier, application tier, data tier) and develop multi-tier applications to understand and experiment with the deployment of enterprise applications.
6. Develop web applications using a combination of client-side (JavaScript, HTML, XML, WML) and server-side technologies (ASP.NET, ADO.NET).

Detailed contents:

- 1. The .NET framework:** Introduction, Common Language Runtime, Common Type System, Common Language Specification, The Base Class Library, The .NET class library Intermediate language, Just-in-Time compilation, garbage collection, Application installation & Assemblies, Web Services, Unified classes. (8 Lectures)
- 2. C# Basics:** Introduction, Data Types, Identifiers, variables & constants, C# statements, Object Oriented Concept, Object and Classes, Arrays and Strings, System Collections, Delegates and Events, Indexes Attributes, versioning. (8 Lectures)
- 3. C# Using Libraries:** Namespace-System, Input Output, Multi-Threading, Networking and Sockets, DataHandling, Windows Forms. (8 Lectures)
- 4. C# in Web application:** Error Handling. (8 Lectures)
- 5. Advanced Features Using C#:** Web Services, Windows services, messaging, Reflection, COM and C#, Localization. Advanced Features Using C#: Distributed Application in C#, XML and C#, Unsafe Mode, Graphical Device Interface with C#, Case Study (Messenger Application) (8 Lectures)

References:

1. Jeffrey Richter, "Applied Microsoft .Net Framework Programming", (Microsoft)
2. Fergal Grimes, "Microsoft .Net for Programmers", (SPD)
3. TonyBaer, Jan D. Narkiewicz, Kent Tegels, Chandu Thota, Neil Whitlow, "Understanding the .Net Framework", (SPD)
4. Balagurusamy, "Programming with C#", TMH

Course Name: Data Communication & Networks

Course Code: MCAIT302

Course Type: Theory	Course Details: IT- 11	L-T-P: 2 - 1- 0	
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5. Identify the different types of network devices and their functions within a network
6. Understand and building the skills of subnetting and routing mechanisms.
7. Familiarity with the basic protocols of computer networks

Detailed contents:

- 1. Data Communication:** History of Data Communication; Importance of Modulation & Multiplexing: Types of Modulation; Multiplexing techniques; The electrical interface-transmission media, attenuation and distortion sources, Signal types and signal propagation delay, public carrier service (5 Lectures)
- 2. Data Transmission basics:** Synchronous/asynchronous transmission; Error detection & Error correction methods; Data compression; Transmission control circuits; Communication control devices; Protocol basics; data link control protocol, Broad band multi service networks- networking requirement, FDDI II, Cell based networks, ATM LAN, DQDB, ATMR, CRMA II (6 Lectures)
- 3. Computer Networks:** Advantages & Disadvantages; Classification of networks; Wired/Wireless networks; protocols, CSMA/CD & CSMA/CA; Ethernet switching – Thinnet & Thicknet; Fast Ethernet; FDDI; Network devices- Active & Passive; WAN-characteristic of public data networks; packet switched; circuit switched; ISDN;private networks; Layered approach to network design: ISO/OSI & TCP/IP models; TCP & UDP; Protocols at each layer; Classful and Classless addressing; Subnetting concept; DNS; Introduction to NAT; IPv4 vs IPv6. (12 Lectures)
- 4. Internetworking:** Inter network architecture issues, IP standards, Routing – Basics, Link State & Distance Vector; ISO routing protocols, Transport Protocols, Application support protocols, application specific protocols (8 Lectures)
- 5. System aspects-**Layer interactions, implementation issues, related standards, Design of Campus-wide LAN; Design of WAN. (8 Lectures)

References:

1. Data and Computer Communication- W.Stallings
2. Computer Networks- A.S.Tanenbaum
3. Computer Network and Distributed Data Processing- J. Martin
4. Local Networks - W.Stallings
5. Computer Communication Network Design and Analysis- M.Schwartz
6. Data Communications and Networking- B. A. Forouzan
7. TCP/IP fundamentals-Vol I & II - Comer

Course Name: Information Retrieval & Web Mining

Course Code: MCAIT303

Course Type: Theory	Course Details: IT- 12		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

On completion of the course students will be expected to:

1. gain an understanding of the basic concepts and techniques in Information Retrieval.
2. understand how statistical models of text can be used to solve problems in IR, with a focus on how the vector-space model and language models are implemented and applied to document retrieval problems;
3. understand how statistical models of text can be used for other IR applications, for example clustering and news aggregation;
4. appreciate the importance of data structures, such as an index, to allow efficient access to the information in large bodies of text;
5. understand common text compression algorithms and their role in the efficient building and storage of inverted indices.

Detailed contents:

- 1. Introduction and Information Retrieval Models:** Introduction to information storage and retrieval, IR models, functional view of paradigm IR system, IR and other types of information systems.
IR evaluation: Measuring effectiveness of IR system; Precision and recall, User centered evaluation. Boolean model, Vector space model, Probabilistic model, non-classical models of IR – Information logic model, cluster model, LSI model; AI in IR – ANN model, genetic algorithm model, Knowledge-bases, Natural language processing; Query expansion. (8 Lectures)
- 2. Data structures and algorithms related to IR:** Data structures -inverted files and its implementation, B- trees and tries, signature files; algorithms – indexing and retrieval algorithms. (8 Lectures)
- 3. Term and query operation:** Tokenization, stop lists, stemming, different types of stemmers; thesaurus construction; query modification. (8 Lectures)
- 4. Vector space model:** Indexing - document and query representation; Term weighting; Similarity measures, ranking algorithms; Query expansion- Relevance feedback methodology Rocchio's and Ide's method, Evaluation of relevance feedback; clustering algorithms and its application. (8 Lectures)
- 5. Further topics:** Web information indexing; Web Information categorization and ranking; Web IR evaluation; Data fusion, Text mining; Semantic Web; Intelligent agents in IR. (8 Lectures)

References:

1. Modern Information Retrieval, Ricardo – Baeza Yates, B. Ribeiro-Neto (Addison Wesley Longman)
2. C. J. van Rijsbergen "Information Retrieval", Second Edition
3. G. Salton, and M. J. McGill, "Introduction to modern information retrieval". New York: McGraw-Hill
4. Information Retrieval: Data Structures and algorithms. W. Frakes and R. Baeza-Yates (Eds.). Prentice Hall
5. D. Jufrasky, and J. H. Martin. "Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition". Pearson education, Asia, 2000.
6. Introduction to Information Retrieval, by C. Manning, P. Raghavan, and H. Schütze

Course Name: Machine Learning Techniques

Course Code: MCAIT304

Course Type: Theory	Course Details: IT- 13		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able:

1. Develop an appreciation for what is involved in Learning models from data
2. Understand a wide variety of learning algorithms
3. Understand how to evaluate models generated from data
4. Apply the algorithms to a real problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models

Detailed contents:

- 1. Introduction:** Difference between human learning and machine learning; Types of Machine learning- Supervised, Unsupervised, Reinforced and their comparison; Applications of Machine learning; Tools for Machine learning; Issues in Machine learning; Basic types of data in Machine learning; Data Pre-processing; Other types of learning – Representation learning, Active learning, Memory based learning, Associative rule learning algorithms, Ensemble learning algorithm (10 Lectures)
- 2. Modelling & Evaluation:** Selecting a model – Predictive and Descriptive models; Training a model for supervised learning; Model representation and Interpretability; Evaluating the performance of a model – Classification, Regression & Clustering; Improving the performance of a model; Basics of Feature Engineering- Feature transformation, Feature Subset selection; Brief overview of Probability – Random variables, Introduction to Discrete distributions and continuous distribution, Central limit theorem, Sampling distributions (10 Lectures)
- 3. Different learning methods:** Bayesian method – Baye’s theorem, Brute force Bayesian algorithm, Baye’s optimal classifier, Bayesian belief network; Supervised learning – Classification, Common classification algorithms – kNN, Decision tree, Random Forest, Support vector machine; Regression – Common regression algorithms, Logistic regression, Maximum likelihood estimation. (6 Lectures)
- 4. Unsupervised learning:** Unsupervised vs Supervised learning; Applications of Unsupervised learning; Clustering – Different types of clustering techniques, Partitioning methods, K medoids, Hierarchical clustering, Density based methods; Finding pattern using association rule; The apriori algorithm for association-based learning. (6 Lectures)
- 5. Basics of Neural Networks:** Understanding the biological neuron; Exploring the artificial neuron; Types of activation functions – Identity function, Threshold function, ReLU function, Sigmoid function, Hyperbolic tangent function; Implementations of ANN; Architectures of Neural Network – Single layer & Multi layer feed forward network, Competitive network, Recurrent network; Learning Process in ANN – Number of layers, Direction of Signal flow, Number of nodes in layers, Weight of interconnection between neurons; Back propagation; Introduction to Deep Learning. (8 Lectures)

References:

1. Miroslav Kubat: An introduction to machine Learning, Springer
2. John Slavo: Machine Learning for Beginners
3. Rajiv Chopra: Machine Learning, Khanna Publications
4. Gareth Witten, James Daniela et al: An introduction to Statistical Learning with applications in R
5. Ethen Alpaydin: Introduction to Machine Learning, PHI
6. Tom M Mitchell: Machine Learning

Course Name: Multimedia Systems

Course Code: MCAIT305

Course Type: Theory	Course Details: IT- 14		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able:

1. describe different realisations of multimedia tools and the way in which they are used
2. analyse the structure of the tools in the light of low-level constraints imposed by the adoption of various QoS schemes (ie bottom up approach)
3. analyse the effects of scale and use on both presentation and lower-level requirements (ie top down approach)
4. plan experiments to test user perception of multimedia tools
5. state the properties of different media streams; compare and contrast different multicast protocols
6. describe mechanisms for providing QoS guarantees in the network and to propose experiments to analyse their performance.

Detailed contents:

- 1. Multimedia Technology:** Meaning & scope of Multimedia; Elements of Multimedia; Creating multimedia applications; Multimedia file & I/O functions; Multimedia data structures; Multimedia file formats; Multimedia Protocols. (8 Lectures)
- 2. Multimedia Audio:** Digital sound; Audio compression & decompression; Companding; ADPCM compression; MPEG audio compression; True Speech; Special effects and Digital Signal Processing; Audio synthesis; FM synthesis; Sound blaster card; Special effect processors on sound cards; Wave table synthesis; MIDI functions; Speech synthesis & Recognition. (8 Lectures)
- 3. Multimedia Video:** Representation of Digital video; Video capture; Frame grabbing; Full motion video; Live video in a window; Video processor; Video compression & decompression; Standards for video compression & decompression; Playback acceleration methods. (8 Lectures)
- 4. Creating Multimedia Animation and Authoring Tools:** Icon animation; Bit-map animation; Real-time vs Frame by Frame animation; Object modeling in 3D animation; Motion control in 3D animation; Transparency; Texture, Shadows, Anti-aliasing; Human modeling & Animation; Automatic motion control. Multimedia Authoring tools: Project editor; Topic editor; Hot-spot editor; Developing a multimedia title; Multimedia text authoring systems; Usage of authoring tools. (8 Lectures)
- 5. Multimedia communication systems:** Multimedia DBMS; Documents, Hypertext and MHEG; User Interfaces; Synchronization, A reference model for multimedia synchronization. Multimedia on LANs & Internet: Fast modems & Digital networks for multimedia; High speed digital networks; Video conferencing techniques; Multimedia interactive applications on internet; Future Directions. (8 Lectures)

References:

1. Multimedia: Computing, Communications & Applications – Nahrstedt & Steinmetz
2. Computer Speech Processing – Fallside F.
3. Speech Analysis, Synthesis & Perception – Flanagan, J.L.
4. Hypertext & Hypermedia- Nielsen J.

Course Name: MAIN PROJECT (Phase I):

Course Code: MCAITP301

Course Type: Practical	Course Details: ITP-1		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		30	20

Course Name: MAIN PROJECT (Phase I): Viva-voce

Course Code: MCAITP302

Course Type: Practical	Course Details: ITP-2		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		30	20

Semester – IV

Course Name: Secure Computing

Course Code: MCAIT401

Course Type: Theory	Course Details: IT- 15		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able:

1. Protect and defend computer systems and networks from cybersecurity attacks.
2. Diagnose and investigate cybersecurity events or crimes related to computer systems and digital evidence.
3. Effectively communicate in a professional setting to address information security issues.

Detailed contents:

- 1. Basics of computer security:** Kinds of security branches, Security goals and vulnerabilities, Methods of Defence (8 Lectures)
- 2. Program Security:** Viruses and other malicious codes, Targeted malicious codes, Controls against program threats; Protection in General purpose operating systems, protecting memory and addressing, Protecting access to general objects, File protection mechanisms, User authentication. (8 Lectures)
- 3. Security in Networks and Distributed Systems:** Threats in Network, Network security control, Administering Security (8 Lectures)
- 4. Encryption and Decryption:** Mono alphabetic substitution ciphers, Poly alphabetic substitution ciphers, Transpositions, Stream and block ciphers, Secure encryption system: Public key Encryption, Merkel Hellman, knapsacks, RSA Encryption, EL Gamel algorithm, Hash Algorithms, Digital Signature algorithm, Symmetric system, DES algorithm, Enhancing cryptographic security. (8 Lectures)
- 5. Legal and Ethical issues in Computer Security:** Protecting programs and data, computer crime, Ethical issues, electronic privacy. (8 Lectures)

References:

1. Security in Computing by Charles P. Pfleeger (Prentice Hall)
2. Computer Security Handbook Vol. 1 & 2 by Bosworth, Kabay & Whyne (Wiley)
3. Applied Cryptograph by Schneier, Bruce (Wiley)
4. Practical Cryptography by Schneier & Ferguson (Wiley)
5. Introduction to cryptography: Principles and applications by Delfs & Knebl (Springer)

Course Name: Embedded System Design

Course Code: MCAIT402

Course Type: Theory	Course Details: IT- 16		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able:

1. Acquire a basic knowledge about programming and system control to perform a specific task.
2. Acquire knowledge about devices and buses used in embedded networking
3. Develop programming skills in embedded systems for various applications.
4. Acquire knowledge about basic concepts of circuit emulators.
5. Acquire knowledge about Life cycle of embedded design and its testing.

Detailed contents:

1. **Classification of systems:** Real-time design issues; Typical Real-time applications; Hard vs Soft real-time systems; A reference model for real-time systems; Design of data acquisition System; Language issues; Exception handling; Characteristics of real-time OS, Real-time kernels; Introduction to iRMX III+, TRON; Inter- task communication & synchronization; Real-time memory management; System performance issues & optimization
Real-time specification & design techniques: Mathematical specification; Structure charts; Pseudo-code and programming design languages; DFDs, Petri Nets, State charts, Usage of Graphical Techniques (8 Lectures)
2. **Commonly used approaches to real-time scheduling:** Clock driven, priority driven; Priority driven scheduling of periodic tasks, Scheduling a-periodic and sporadic jobs in priority driven systems; Resources and resource access control; Multiprocessor scheduling, resource access control & synchronization
Real-time kernels: Polled loop systems; State driven code; Co-routines; Interrupt driven systems- Context Switching, Round Robin systems, Pre-emptive priority systems; Major & Minor processing; Foreground/Background systems; Full featured real-time OS (8 Lectures)
3. **Inter-task communication & Synchronization:** Buffering data- Time-relative buffering, Ring Buffers; Mailboxes & their implementation, Other operation on Mailboxes; Queues; Critical regions; Semaphores- Counting Semaphores, Problems with semaphores; Deadlock- Detection & Recover, Avoidance Real-Time Memory Management: Process Stack Management; Dynamic Allocation; Static schemes. (8 Lectures)
4. **System performance analysis & optimization:** response-time calculation; Interrupt Latency; Time loading and its measurement; Reducing response times and time loading; Analysis of Memory requirements; Reducing Memory loading. (8 Lectures)
5. **Comparative study of real-time programming languages;** Design of Embedded Systems – Techniques & issues involved. (8 Lectures)

References:

1. Automatic Control Systems by Kuo, B.C.
2. Real-time Computer Control by Linkens & Bennett
3. Real-time software for small systems by Leigh, A.W.
4. Programming embedded microprocessors by Fowler, R.J.
5. Real-time systems Design & Analysis by Laplante P.A.

Course Name: Digital Communication

Course Code: MCAIT403

Course Type: Theory	Course Details: IT- 17		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

On completion of the course, student will be able to

1. Understand working of waveform coding techniques and analyse their performance.
2. Analyze the performance of a baseband and pass band digital communication system in terms of error rate and spectral efficiency.
3. Perform the time and frequency domain analysis of the signals in a digital communication system.
4. Design of digital communication system.
Understand working of spread spectrum communication system and analyze its performance.

Detailed contents:

- 1. Elements of Digital Communication Systems:** Elements of Digital Communication Systems: Model of Digital Communication Systems, Digital Representation of Analog Signal, Certain issues in Digital Transmission, Advantages of Digital Communication Systems, Bandwidth-S/N trade off, Hartley Shannon Law, Sampling Theorem. (5 Lectures)
- 2. Pulse Code Modulation:** Pulse Code Modulation: PCM Generation and Reconstruction, Quantization noise, Nonuniform Quantization and Companding, DPCM, Adaptive DPCM, DM and Adaptive DM. Noise in PCM and DM. (5 Lectures)
- 3. Digital Modulation Techniques:** Digital Modulation Techniques: Introduction, ASK, ASK Modulator, Coherent ASK Detector, Non-Coherent ASK Detector, FSK, Bandwidth and Frequency Spectrum of FSK. Non coherent FSK Detector, Coherent FSK Detector, FSK Detection Using PLL, BPSK, Coherent PSK Detection, QPSK, Differential PSK. (5 Lectures)
- 4. Baseband transmission and Optimal Reception of Digital Signal:** Baseband transmission and Optimal Reception of Digital Signal: Pulse shaping for optimum transmissions. A Baseband Signal Receiver, Probability of Error. Optimum Receiver, optimum of Coherent Reception. Signal Space Representation and Probability of Error, eye diagrams, Cross talk. (5 Lectures)
- 5. Information Theory:** Information Theory: Information and entropy, conditional entropy and redundancy, Shannon Fano coding, Mutual Information, Information loss due to noise, source coding - Huffman Code, variablelength coding, Source coding to Increase average Information per bit; Lossy source coding. (5 Lectures)
- 6. Linear Block Codes:** Linear Block Codes: Matrix description of Linear Block Codes, Error detection and error Correction capabilities of linear block codes; Cyclic Codes: Algebraic structure, encoding, syndrome calculation. Decoding (5 Lectures)
- 7. Convolution Codes:** Encoding; Decoding using State, tree and trellis diagrams. Decoding using Viterbi algorithm; Comparison of Error Rates in Coded and Uncoded Transmission. (5 Lectures)
- 8. Spread Spectrum Modulation:** Spread Spectrum Modulation: Use of Spread Spectrum, Direct Sequence Spread Spectrum (DSSS), Code Division Multiple Access, Ranging using DSSS. Frequency Hopping Spread

Spectrum, PN - sequences: Generation and Characteristics. Synchronization in Spread Spectrum Systems (5 Lectures)

References:

1. Principles of communication systems - Herbert Taub. Donald L Schiling, Goutam Sana, 3rd Edition, McGraw-Hill,2008.
2. Digital and Analog Communicator Systems - Sam Shanmugam, John Wiley, 2005.
3. Digital Communications - John G. Proakis . Masoud salehi – 5th Edition, McGraw-Hill, 2008.
4. Digital Communication - Simon Haykin, Jon Wiley, 2005.
5. Digital Communications - Ian A. Glover, Peter M. Grant, Edition, Pearson Edu., 2008.
6. Communication Systems-B.P. Lathi, BS Publication, 2006.

Course Name: Image Processing

Course Code: MCAIT404

Course Type: Theory	Course Details: IT- 18		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to

1. understand the need for image transforms different types of image transforms and their properties.
2. develop any image processing application.
3. learn different techniques employed for the enhancement of images.
4. learn different causes for image degradation and overview of image restoration techniques.
5. understand the need for image compression and to learn the spatial and frequency domain techniques of image compression.
6. learn different feature extraction techniques for image analysis and recognition

Detailed contents:

- 1. Introduction:** Image representation and modeling, 2-D linear system, Luminance, Contrast and Brightness,Color representation, Visibility functions, Monochrome and color vision model. (8 Lectures)
- 2. Image Quantization and Image Transforms:** Sampling theorem, Anti-aliasing, image quantization, Orthogonal and unitary transforms, DFT, Cosine transform, Hadamard transform, Haar transform, KL transform. (8 Lectures)
- 3. Image Enhancement:** Point operation, Histogram modeling, Filtering and spatial operations, Transformoperations, Multi-spectral Image Enhancement. (8 Lectures)
- 4. Image Restoration:** Image formation models, Noise models, Inverse and Wiener filtering, Least square filters, Recursive filters, Maximum entropy method, Blind de-convolution, Bayesian method of noise removal, Image reconstruction, Tomography, Radan transform, Back-projection, Reconstruction algorithm, Algebraic method ofreconstruction, Fan-beam reconstruction. (8 Lectures)
- 5. Data Compression:** Data compression vs. Bandwidth, Pixel coding, Predictive coding, Transform coding,Coding of two-tone images. (8 Lectures)

References:

1. Fundamentals of Digital Image Processing: Anil K. Jain
2. Digital Image Processing: R. Chellappa
3. Image Processing for Scientific Applications: Bernd Jahne
4. Digital Image Processing: R.C. Gonzalez & R.E. Woods
5. The Image Processing Handbook: J.C. Russ
6. Digital Image Processing: W.K. Pratt
7. Digital Image Restoration: Andrews & Hunt

Course Name: Data Science & Python/R**Course Code: MCAITE401**

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to

1. develop relevant programming abilities.
2. demonstrate proficiency with statistical analysis of data.
3. develop the ability to build and assess data-based models.
4. execute statistical analyses with professional statistical software.
5. demonstrate skill in data management.
6. apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

Detailed contents:

- 1. Introduction to Data Science with Python/R:** What is analytics & Data Science, Common Terms in Analytics, Analytics vs. Data warehousing, OLAP, MIS Reporting, Relevance in industry and need of the hour, Types of problems and business objectives in various industries, How leading companies are harnessing the power of analytics, Critical success drivers, Overview of analytics tools & their popularity, Analytics Methodology& problem solving framework, List of steps in Analytics projects, Identify the most appropriate solution design for the given problem statement, Project plan for Analytics project & key milestones based on effort estimates, Build Resource plan for analytics project, Why Python for data science, Overview of Python- Starting with Python. (8 Lectures)
- 2. Python Essentials:** Introduction to installation of Python, Introduction to Python Editors & IDE's(Canopy, pycharm, Jupyter, Rodeo, Ipython etc), Understand Jupyter notebook & Customize Settings, Concept of Packages/Libraries - Important packages (NumPy, SciPy, scikit-learn, Pandas, Matplotlib, etc), Installing & loading Packages & Name Spaces, Data Types & Data objects/structures (strings, Tuples, Lists, Dictionaries), List and Dictionary Comprehensions, Variable & Value Labels – Date & Time Values, Basic Operations - Mathematical - string – date, Reading and writing data, Simple plotting, Control flow & conditional statements, Debugging & Code profiling, How to create class and modules and how to call them, Importing Data from various sources (Csv, txt, excel, access etc). (8 Lectures)
- 3. Accessing/Importing and Exporting Data using Python Modules:** Database Input (Connecting to database), Viewing Data objects - subsetting, methods, Exporting Data to various formats, Important python modules: Pandas, beautiful soup, Cleansing Data with Python, Data Manipulation steps (Sorting,

filtering, duplicates, merging, appending, subsetting, derived variables, sampling, Data type conversions, renaming, formatting etc),Data manipulation tools(Operators, Functions, Packages, control structures, Loops, arrays etc),Python Built-in Functions (Text, numeric, date, utility functions),Python User Defined Functions, Stripping out extraneous information, Normalizing data, Formatting data, Important Python modules for data manipulation (Pandas, Numpy, re, math, string, date, time etc). (8 Lectures)

- 4. Data Analysis:** Visualization using Python/R-introduction exploratory data analysis; Descriptive statistics, Frequency Tables and summarization, Univariate Analysis (Distribution of data & Graphical Analysis), Bivariate Analysis(Cross Tabs, Distributions & Relationships, Graphical Analysis),Creating Graphs-Bar/pie/line chart/histogram/ boxplot/ scatter/ density etc),Important Packages for Exploratory Analysis (NumPy Arrays, Matplotlib, seaborn, Pandas and scipy.stats etc),Basic Statistics - Measures of Central Tendencies and Variance,Building blocks - Probability Distributions - Normal distribution - Central Limit Theorem, Inferential Statistics - Sampling - Concept of Hypothesis Testing, Statistical Methods - Z/t-tests(One sample, independent, paired), Anova, Correlations and Chi-square, Important modules for statistical methods: Numpy, Scipy, Pandas. (8 Lectures)
- 5. Text Mining and Analytics:** Sentiment Analysis using Python, Text Analytics – Word cloud analysis using Python, Text Analytics - Segmentation using K-Means/Hierarchical Clustering, Text Analytics - Classification (Spam/Not spam),Applications of Social Media Analytics, Metrics(Measures Actions) in social media analytics, Examples & Actionable Insights using Social Media Analytics, Important python modules for Machine Learning (SciKit Learn, stats models, scipy, nltk etc),Fine tuning the models using Hyper parameters, grid search, pipingetc. (8 Lectures)

References:

1. Getting started with Data Science: Making sense of Data with Analytics Paperback 2016 by Murtaza HaiderKindle edition (2015) IBM Press.
2. Statistics for Data Science Paperback 2017 by James . D.Miller Kindle edition.
3. Data Science from Scratch Paperback –i January 2015 by Joel Grus (OREILLY)
4. Pandas for Everyone: Python Data Analysis 1e Paperback-15 April 2018 BY Daniel.Y.Chen.
5. Hands on Data Analysis with Numpy and Pandas Paperback -29 June 2018 by Curtis Miller–Kindle edition.
6. Python for Data Analysis: Data Wrangling with Pandas, Numpy and Python paper back 27 October 2017 byWillima Mckinney, Kindle Edition.

Course Name: Robotics & AI

Course Code: MCAITE402

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to

1. Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
2. Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
3. Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.
4. explain the fundamentals of robotics and its components
5. illustrate the Kinematics and Dynamics of robotics

6. elucidate the need and implementation of related Instrumentation & control in robotics
7. illustrate the movement of robotic joints with computers/microcontrollers.
8. Explain sensors and instrumentation in robotics

Detailed contents:

- 1. Introduction:** Scope of AI: Games, theorem proving, Natural language processing, Vision and speech processing, Robotics & Expert systems, AI techniques, Introduction to intelligent agents.
Search Techniques: State space search, control strategies: Depth first search, Breadth first search and Production systems; Use of heuristics: Hill climbing, Best first search, A* algorithm- admissibility, AND/OR graph– AO*, Constraint satisfaction; Game playing: Minimax and Alpha-Beta searching, Genetic algorithms. (12 Lectures)
- 2. Knowledge Representation:** Propositional logic: its syntax and semantics; Reasoning patterns in propositional logic: Resolution, forward and backward reasoning. First order logic: Syntax and semantics; Inference in first order logic: Unification, Forward & backward chaining, Resolution. Structured knowledge representation: Semantic Net, Frames, and Conceptual graphs; Uncertain knowledge and reasoning: Introduction to probabilistic reasoning; representing vagueness- fuzzy sets and fuzzy logic.
Learning: Different forms of learning; Concept learning system; Inductive learning; Learning decision trees; Neural network: single layer feed forward network. (10 Lectures)
- 3. Introduction to Robotics:** Classification; Robot manipulator kinematics, Arm control & Wrist control equations; Trajectory generation & control; Task planning & compliance; Linear & Non-linear control of manipulators; Robot vision. (8 Lectures)
- 4. Robot Motion:** Translational & Rotational motion; Motion conversion; Lagrangian analysis of manipulator; Control of actuators; Robot sensory devices. (5 Lectures)
- 5. Computer Vision & Robot Programming:** Imaging components; Object recognition & categorization; Need for vision training & adaptation; Robot Programming; Robot design & Process specifications. (5 Lectures)

References:

1. Artificial Intelligence: Rich and Knight
2. Artificial Intelligence: A Modern Approach: Stuart Russell and Peter Norvig
3. Introduction to Artificial Intelligence: Partick Winston
4. Artificial Intelligence: Nilsson
5. Introduction to Robotics: Fu, Lee & Gonzalez
6. Introduction to Robotics- Mechanics & Control: Craig, John J
7. Robot Motion: Brady
8. Robot Manipulators, Mathematics, Programming & Control: Paul R.
9. Robot Modelling: Ho C Y

Course Name: Distributed Computing

Course Code: MCAITE403

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

On successful completion of this course, the student should:

1. Have an understanding of the fundamental issues in distributed computing sufficient to form a sound basis for the design and implementation of distributed systems from scratch.
2. Be aware of the problems and pitfalls relating to the operation of distributed systems, and have insight into the main techniques for addressing them.

Detailed contents:

- 1. Introduction to Distributed Systems:** Goals and advantages of distributed systems, Distributed Computing System Models, Network operating systems, True distributed systems, Design issues – Transparency, Reliability, Performance and Scalability; Message Passing, Group communication. (8 Lectures)
- 2. RPCs and Distributed Shared Memory:** Remote procedure call (RPC), Light Weight RPC, Client Server Communication. Introduction to shared memory, Comparison of shared memory systems, Consistency models, Page-based distributed shared memory. (8 Lectures)
- 3. Synchronization in Distributed Systems:** Clock synchronization, Logical and physical clocks, Vector clock, Clock synchronization algorithms, Mutual exclusion – Centralized, Distributed, Token Ring and Maekawa’s Algorithm, Election Algorithms, Atomic transactions, Deadlocks in distributed systems. (8 Lectures)
- 4. Processes and Processors in Distributed Systems:** Process Migration, Threads – Usage, Design issues and implementation, Scheduling in distributed systems, Load balancing and Load sharing, Fault tolerance. (8 Lectures)
- 5. Distributed File Systems and Security:** File Models, File Accessing Models, File Sharing Semantics, File Caching Schemes, File Replication. Security in Distributed System: Potential Attacks to Computer Systems, Introduction to Cryptography, Authentication, Access Control; Case Studies: V -Systems, Amoeba. (8 Lectures)

References:

1. Distributed Operating Systems: A.S. Tanenbaum
2. Distributed Systems Concepts and Design: G.F. Coulouris, J.Dollimore and T. Kindberg
3. Distributed Operating Systems: Pradeep K. Sinha

Course Name: Cloud Computing

Course Code: MCAITE404

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to

1. Understand the concepts, characteristics, delivery models and benefits of cloud computing
2. Understand the key security and compliance challenges of cloud computing
3. Understand the key technical and organisational challenges
4. Understand the different characteristics of public, private and hybrid cloud deployment models.

Detailed contents:

- 1. Introduction:** Understanding the Cloud Computing landscape; From collaborative to the Cloud – A short history Client – Server Computing, Peer-to-Peer Computing, Distributed Computing, Collaborative Computing, Cloud Computing, Functioning of Cloud Computing, Cloud Architecture, Cloud Storage, Cloud based Service Offerings & applications; Real cases and applications of Cloud Computing -Service Modeling, Infrastructure Services, Platform Services, Software Services - Software as service modes- Massively scaled software as a service- Scale of Economy, Management and Administration. (8 Lectures)
- 2. Inside Cloud Computing:** Making Strategy Decisions- Governance Issues- Monitoring Business Processes; Cloud Service Administration- Service Level Agreements and Monitoring-Support Services- Accounting Services, Resource Management- IT Security- Performance Management- Provisioning- Service Management, Untangling Software Dependencies. (8 Lectures)
- 3. Cloud Computing Technology & Storage:** Clients – Mobile – Thin – Thick, Security - Data Linkage, Network- Basic Public Internet- The Accelerated Internet- Optimized Internet Overlay- Site-to-Site VPN- Cloud Providers, Cloud Consumers, Bundling instances; Elastic Block storage and its usage; Simple DB Service, NoSQL database; Storage security; Merits and demerits of storage. (8 Lectures)
- 4. Accessing the Cloud & Data Management:** Platforms- Web Application Framework- Web Hosting Services- Proprietary Methods, Web Applications- API's in Cloud Computing, Browsers for Cloud Computing; Amazon webservices and its integration; Introduction to Microsoft Cloud and its use in .NET environment; Data Security- Data Location- Data Control- Securing data for transport, Scalability and Cloud Services- Large Scale Data Processing- Databases and Data Stores- Data Archival. (8 Lectures)
- 5. Discovery of Private and Hybrid Clouds:** Defining a private cloud- Public, Private, and Hybrid Clouds – A Comparison, Examining the Economics of the private cloud- Assessing capital expenditures- Vendor Private Cloud Offerings, The Up Key Vendors- Service Oriented- Systems Integrators- Technology Enablers; Cloud Computing standards; Cloud Governance; Basics of Distributed computing: Basics of Hadoop; Hadoop Distributed File System, Introduction to HIVE, OPSWorks Application management Service; Migrating to the cloud using the WAVE approach. (8 Lectures)

References:

1. Bloor R, Kaufman M, Halper F, Judith Hurwitz: Cloud Computing for Dummies, Wiley India
2. Anthy T Volte: Cloud Computing – A Practical approach, McGraw Hill
3. James E Smith, Ravi Nair: Virtual Machine, Morgan Kaufmann
4. John Rittinghouse & James Ransome: Cloud Computing implementation management & strategy, CRC Press
5. Michael Miller: Cloud Computing- Web based application that changes the way you work and collaborate online, Que Publishers
6. Syed A Ahson, Mohd Ilyas: Cloud Computing and Software Services – Theory & Techniques, CRC Press

Course Name: Mobile Computing & Applications

Course Code: MCAITE405

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. Define mobile technologies in terms of hardware, software, and communications.

2. Utilize mobile computing nomenclature to describe and analyze existing mobile computing frameworks and architectures.
3. Evaluate the effectiveness of different mobile computing frameworks.
4. Describe how mobile technology functions to enable other computing technologies.

Detailed contents:

1. **Introduction:** Evolution and fundamentals of Mobile communication, Terminal mobility, Personal Mobility and Service Portability, The Cellular concept, A basic cellular system, Multiple access technologies of cellular systems, Analog & Digital cellular systems, Generations of systems, Cellular system operation and planning, System architecture, Location updating and call setup, Handoff & power control. (8 Lectures)
2. **Digital cellular systems:** Evolution of Mobile System, GSM, GSM standardization and service aspects, GSM reference architecture and function partitioning, GSM radio aspects, GSM security aspects, GSM protocol model, GSM call flow sequences, gprs, edge; (Wireless) Medium Access Control: Motivation for a specialized MAC (Hidden and exposed terminals, Near and Far terminals), SDMA, FDMA, TDMA, CDMA. (8 Lectures)
3. **Mobile network and transport layer:** Mobile IP: goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, registration, tunneling and encapsulation, optimizations, DynamicHost Configuration Protocol (DHCP). Mobile Transport Layer: Traditional TCP ,Indirect TCP ,Snooping TCP, Mobile TCP, Fast retransmit/fast recovery ,Transmission /time-out freezing ,Selective retransmission, Transaction oriented TCP. (8 Lectures)
4. **Database issues:** Hoarding techniques, caching invalidation mechanisms. client server computing with adapt ion, power-aware and context-aware computing, transactional models, query processing, recovery, and qualityof service issues. (8 Lectures)
5. **Mobile Ad hoc Networks (MANETs):** Overview, Properties of a MANET, spectrum of MANET, applications, routing and various routing algorithms, security in MANET's. (8 Lectures)

References:

1. Mobile and Personal communication systems and Services – Raj Pandya
2. Mobile Communications – Schiller J.
3. Mobile Cellular Telecommunications – Lee William C.Y.
4. Wireless Communications & Networks – Stallings, William

Course Name: Big Data Analysis

Course Code: MCAITE406

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. To study the basic technologies that forms the foundations of Big Data.
2. To understand the specialized aspects of big data including big data application, and big data analytics.
3. To study different types Case studies on the current research and applications of the Hadoop and big data in industry
4. To study Data warehousing and OLAP using some real-life examples

Detailed contents:

- 1. Introduction:** Big Data Overview, characteristics, architecture, technologies, applications. Cloud Computing- Concept. Comparative study of working of various cloud services platforms – Amazon Web Services, Microsoft Azure, Google Cloud Platform. Different types of cloud computing services. (8 Lectures)
- 2. Data Analysis and Visualization:** Analysis - The process of data analysis-requirement, collection, processing, cleaning, exploratory data analysis, modeling and algorithms, data product, communication. Visualization - Characteristics of effective graphical displays. (8 Lectures)
- 3. Statistical Concepts in Big Data Analysis:** Probability theory, probability distribution, statistical significance, hypothesis testing and regression. (8 Lectures)
- 4. Overview of Hadoop and its Ecosystem:** HDFS, MapReduce, Pig, Hive, HBase, Spark, Sqoop, Oozie. (8 Lectures)
- 5. NoSQL Databases:** Column oriented databases (Cassandra, HBase), Document oriented database (MongoDB), Key-Value pairs (CouchBase), Graph database (Neo4j). (8 Lectures)

References:

1. Big Data For Dummies: Alan Nugent, Fern Halper, Judith Hurwitz, and Marcia Kaufman
2. Hadoop- The Definitive Guide: Tom White, O'Reilly
3. Be A Big D: Deep Pandey
4. Hadoop For Dummies: Dirk deRoos

Course Name: Artificial Neural Networks

Course Code: MCAITE407

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains.
2. Implement deep learning algorithms and solve real-world problems.

Detailed contents:

- 1. Introduction:** Real and artificial neural networks, Neurons as Processing elements, Activation and Signals, Threshold functions, Connection topologies. (5 Lectures)
- 2. Network Models:** Dynamical Systems viewpoint, Additive dynamics, Additive neuronal feedback, Bivalent additive associative memory, Stability of BAMs, Lyapunov functions, Bivalent BAM theorem, BAM connection matrices, Linear associative memory, Optimal linear associative memory, Memory capacity, Hopfield model, Additive dynamics and Noise-Saturation dilemma, Grossberg's saturation theorem, Multiplicative dynamics, Shunting inhibition, Adaptive resonance theory of Grossberg, ART1 and ART2 networks. (15 Lectures)
- 3. Network Learning Methods:** Unsupervised learning, Learning laws, Signal Hebbian, Competitive, Differential Hebbian and differential competitive learning laws, Deterministic and stochastic learning, Stochastic equilibrium, Asymptotic centroid estimation, Kohonen's self-organizing map, Reinforcement learning. (10 Lectures)
- 4. Network Learning Methods:** Supervised learning, Stochastic approximation, Perceptron learning theorem, LMS algorithm, Back-propagation algorithm, Multilayer feed-forward networks, Enhancements in Back-propagation algorithm, Recurrent back-propagation. Adaptive vector quantization, AVQ algorithms, Convergence theorem, Adaptive bi-directional associative memory. (10 Lectures)

References:

1. Neural Networks and Fuzzy Systems: Bart Kosko
2. Neural Computing: Theory and Practice: P.D.Wasserman
3. Artificial Neural systems: P.K.Simpson
4. Neurocomputing: Robert Hecht-Nielson
5. Neural Networks: J.A.Freeman

Course Name: Computer Vision

Course Code: MCAITE408

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. Describe the scope of challenges and applications addressed by computer vision
2. Demonstrate and experiment with image filtering techniques
3. Make use of geometric camera models and multiple view geometry
4. Undertake video analysis problems such as tracking and structure from motion
5. Explain the application of neural networks to computer vision
6. Analyse cognitive tasks including image classification, recognition and detection

Detailed contents:

- 1. Images:** Sources of imagery, The physics of imaging, Representing, acquiring, and displaying images, Grayscale, color, noise, lens distortion, and filtering. (5 Lectures)
- 2. Image processing:** Preprocessing, and image correction, Enhancing features and correcting imperfections, Addressing noise, lens distortion, and blurring. (5 Lectures)
- 3. Computer Vision Paradigms:** Bottom-up, top-down, neural net, feedback, Pixels, lines, boundaries, regions, and object representations, "Low-level", "intermediate-level", and "high-level" vision, Historical and illustrative examples. (5 Lectures)
- 4. Finding Edges and Lines:** Finding edges (low-level), Gradients, zero crossing detectors, line models, Roberts, Sobel, Canny, Finding and grouping lines (intermediate-level), Boundary tracing, line fitting, Hough transform. (5 Lectures)
- 5. Finding and Processing Regions:** Finding "elementary regions" (low-level), Merging, splitting, and grouping regions (intermediate-level). Grouping and analyzing lines and regions (high-level), Guzman, Clowes and Huffman, Waltz. Stereo, and Motion, Optical Flow and FOE, Motion Understanding Texture, Fourier transform Representing the environment and Matching, Clouds, generalized cylinders, semantic nets, Matching line and region groups to object representations (high-level).
Applications: Using computer vision, Applications in Medicine, Industry, and Surveillance. (20 Lectures)

References:

1. Digital Image Processing - R.C.Gonzalez & P.Wintz
2. Computer Vision - D.H.Ballard & C.M.Brown
3. Syntactic Pattern Recognition: An introduction -R.C.Gonzalez and M.G.Thomason
4. Pattern Recognition - A Statistical Approach - P.A. Devijver and J. Kittler
5. Digital Image Processing - W. K. Pratt
6. Fundamentals of Digital Image Processing - A.K. Jain
7. Digital Picture Processing - A. Rosenfeld and A.C. Kak
8. Vision in Man and Machine - M.D. Levine

Course Name: High Performance Computing

Course Code: MCAITE409

Course Type: Theory	Course Details: ITE		L-T-P: 2 - 1- 0
Credit: 3	Full Marks: 50	CA Marks	ESE Marks
		Theoretical	Theoretical
		15	35

Course Learning Outcomes:

Students will be able to:

1. Understand the role of HPC in science and engineering.
2. Be familiar with popular parallel programming paradigms.
3. Understand commonly used HPC platforms with particular reference to Cluster system.
4. Understand the means by which to measure, assess and analyse the performance of HPC applications.

5. Understand the role of administration, workload and resource management in an HPC management software.
6. Understand the mechanisms for evaluating the suitability of different HPC solutions to solving scientific problems

Detailed contents:

1. **Introduction:** Motivation for high performance and parallelism application areas, Technologies, Abstraction levels, Models of Computation; Overheads, Multiple program counters, Multi-threaded execution models; Parallel languages and compilers, Task Parallel and Data parallel Programming models; Memory architecture; Memory access times and associated overheads; Performance timing; Restructuring for parallel performance; Parallelizing compilers; Load & Data transformation; State of the art research & future direction(5 Lectures)
2. **Microprocessor and system architecture:** Pipelining, superscalar designs; SIMD, multithreading; caches;memory, busses; Asynchronous microprocessors for high performance processing and low power applications. (5 Lectures)
3. **Multi-processor architectures:** Classification; programming models; application examples; interconnectionnetworks. (5 Lectures)
4. **Tightly coupled systems:** Cache coherence, consistency, synchronization; SMP, ccNUMA, COMA; performance evaluations. (5 Lectures)
5. **Other architectures and extensions:** Distributed memory systems; vector systems; clusters, grids Supercomputer Architecture; Vector Machines; Parallel Processors; Data Parallel Processors. Single-Instruction- Multiple-Data; Multiple-Instruction-Multiple-Data; Pipelining; Vectorization; Parallelization; Superscalar execution– VLIW computation; PRAMs – parallelism issues Comparison of Serial, Parallel and Vector Architectures. Performance Measures and Models; Speed-up limitations; Theoretical Timings; Efficiencies; Amdahl's Law and Extensions; Scaled Speed-up; Pipeline Speed-up. Data Dependency Reduction. Data flow; Loop reordering. Parallelization of Algorithms; Parallel linear algebra routines; Loop optimizations; Implementation; Principle of Locality; Caches and Buffers. (20 Lectures)

References:

1. J. J. Dongarra, I. B. Duff, D. C. Sorensen and H. A. van der Vorst: Solving Linear Systems on Vector and SharedMemory Computers, SIAM, 1991.
2. K. Hwang: Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill
3. D. A. Patterson and J. L. Hennessy, Computer Architecture: A Quantitative Approach, Morgan Kaufmann
4. D. Kuck: The Structure of Computers and Computations, Wiley
5. Levesque and Williamson: A Guidebook to FORTRAN on Supercomputers, Academic Press
6. Metcalf: FORTRAN Optimization, Academic Press
7. J. M. Ortega: Introduction to Parallel and Vector Solution of Linear Systems, Plenum
8. Quinn: Designing Efficient Algorithms for Parallel Computers, McGraw-Hill
9. P. J. Hatcher and M. J. Quinn: Data-Parallel Programming on MIMD Computers, MIT Press
10. Parallel processor architecture & VLSI Hardware: Decegama A.L.
11. Introduction to Parallel Algorithms & Architectures – Arrays, Trees and Hypercubes: Leighton F. Thomson

Course Name: MAIN PROJECT (Phase II):

Course Code: MCAITP401

Course Type: Practical	Course Details: ITP-3		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 100	CA Marks	ESE Marks
		Theoretical	Theoretical
		60	40

Course Name: MAIN PROJECT (Phase II): Viva-voce

Course Code: MCAITP402

Course Type: Practical	Course Details: ITP-4		L-T-P: 0 - 0 - 4
Credit: 2	Full Marks: 100	CA Marks	ESE Marks
		Theoretical	Theoretical
		60	40