SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE

DEPARTMENT OF COMPUTER ENGINEERING

M. TECH. (Computer Engineering)

Year: 2022-23

Department Vision and Mission

Vision

To become a strong centre of excellence for creating competent human resource in the field of Computer Science and Engineering meeting the dynamic societal and industrial needs.

Mission

M1: To produce technically competent professionals in Computer Science and Engineering having a blend of theoretical knowledge and practical skills.

M2: To encourage innovation, research and analytical activities with professional ethics and responsibilities through quality education.

M3: To provide learning ambience in collaboration with industries to keep pace with dynamic technological advancements and promote spirit of entrepreneurship**.**

M4: To motivate students to apply knowledge to resolve societal and environmental challenges and engage in continuous learning towards sustainable development.

M.Tech. Computer Engineering

PO1 An ability to independently carry out research/investigation and development work to solve practical problems.

PO2 An ability to write and present a substantial technical report/document.

PO3 Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program.

M.Tech. Computer Engineering

PEO1 To prepare post graduates with strong theoretical fundamentals and practical skills to take up technical challenges.

PEO2 To equip post graduates with skill and attitude for research and development towards solving real life socio-economic problems.

PEO3 To develop good team builders with competency to become entrepreneurs in line with national mission of self-reliant India.

M.Tech. Computer Engineering

PSO1 To gain ability to use knowledge to identify research gaps and provide new ideas and innovative solutions to meet needs of industry and society.

PSO2 To be able to take up higher studies, research and development, and entrepreneurships in the contemporary computing environment.

PSO3 To inculcate professional ethics, communication abilities and quest for continuous learning.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A
CO 71016: PROGRAMMING SYSTEMS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basics of Programming.

COURSE OBJECTIVES: To provide a comprehensive skill for problem solving using the programming system.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Develop ability to write a computer program to solve specified problems.
2. Develop skills in algorithmic problem-solving, expressed in a programming language like C, C++, python, Java.
3. Build and document the industry ready program.
4. Solve the real world business problems using programming system.

COURSE CONTENTS:
THEORY:

UNIT 1 Fundamentals of programming systems, elements of programming systems, basic data types and their memory representations, Operators and Expressions, I/O streams, statements, Sequential and conditional execution, Iterative solutions, arrays, matrices and their applications, Strings, Elements of string processing,.

UNIT 2 Introduction to pointers, types of pointers, arithmetic of pointers and use of pointers in applications, functions, types of functions, declaration, definition, scope, parameter passing and recursion, structures, union, pointers to structures, file processing, file creation, I/O operations on files, file functions, working with text and binary files.

UNIT 3 Introduction to Standard Template Library (STL): components of STL: containers, iterators, algorithms, functions, Introduction to container, types of containers and their use, types of Iterators, building application using STL, boots libraries.

UNIT 4 Programming life cycles, coding standards, code tuning techniques, version controlling: Git, CVC etc. Internals of program compilation and execution, Programming errors and error handling. Documentation tools: LaTex, bibtex etc.

UNIT 5 Introduction to object oriented Programming paradigm: object, class, object oriented design principles: encapsulation, information hiding, inheritance, Polymorphism, Overriding & Overloading, Comparison of Procedural and Object Oriented Programming paradigm.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. B. Kernighan & D. Ritchie, " The ANSI C programming Language", PHI, 2000.
2. R.G. Dromey, "How to solve it by computer", PHI,1992.
3. Josuttis, Nicolai M., "C++ Standard Library, The: A Tutorial and Reference", Addison Wesley Longman, Inc, 1999.
4. George Grätzer, "More Math Into LaTeX", 5th edition, Springer, 2016, ISBN 978-3-319-23796.

REFERENCE BOOKS:

1. Timothy Budd, "An Introduction to Object-Oriented Programming", Addison-Wesley Publication, 3rd Edition 2002.
2. YeshwantKanetkar, "Let us C", BPB Publications, 2002.
3. G. Booch, "Object Oriented Analysis & Design", Addison Wesley, 2006
4. Herbert Schildt, "Java 2: The Complete Reference", McGraw-Hill Osborne Media, 11th Edition, 2018.
5. Josuttis, Nicolai M., "C++ Standard Library, The: A Tutorial and Reference", Addison Wesley Longman, Inc, 1999

RESEARCH JOURNALS:

1. ACM Transactions on Programming Languages and Systems
2. Elsevier Journal on Science of Computer Programming

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A
CO71017: MODERN COMPUTER NETWORKS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Programming(C/C++/Java), Linux.

COURSE OBJECTIVES: To enable student in understanding networks, its challenges and various solutions.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Identify errors in a network and able to debug these networking errors.
2. Apply various protocols and algorithms to design and implement a computer network.
3. Estimate the performance of various networking devices and can argue on efficiency related issues.
4. Identify various challenges of the implemented network and can present optimized solutions for those challenges.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to computer networks & their uses, Different topologies. ISO-OSI model: Layered Architecture, Peer-to-Peer processes and encapsulation, Function and Services of OSI layers; The Physical layer: Digital Signals, Transmission Impairments and Maximum data rate of a channel, Shennons theorem, Nyquist theorem. Circuit, Packet and Message switching, virtual Circuit. The data link layer: Design issues & function, Error detection & correction, Hamming code & CRC codes, Framing: Fixed size and Variable size Frame, Bit stuffing and Byte stuffing. Data link layer protocols: Simplest, Stop and Wait, Sliding window protocols, The medium access sublayer: Static and Dynamic Channel Allocation, Protocols: ALOHA Protocol, CSMA (CSMA/CD, CSMA/CA).

UNIT 2 IEEE 802.3, LAN Devices: HUB, Switches- Learning, Cut-Through and store and forward switches, Internetworking Devices: Routers & gateways. The network layer: Design issues and functions, TCP/IP Protocol Architecture: ARP/RARP, IP addressing, IP Datagram format and its Delivery, Routing table format.

UNIT 3 IPv4 fragmentation, Subnet, Supernet, CIDR. Different ICMP messages. Routing algorithms: Shortest path routing, Flooding, LSR, Distance Vector Routing, Hierarchical Routing. Routing Protocols: BGP- Concept of hidden network and autonomous system, An Exterior gateway protocol, Different messages of BGP. Interior Gateway protocol: RIP, OSPF.

UNIT 4 Transport layer: Multiplexing and ports, TCP: Segment format, Sockets, Synchronization, Three Way Hand Shaking, Variable window size and Flow control, Timeout and

Retransmission algorithms, Connection Control, Silly window Syndrome, UDP: Message Encapsulation, Format and Pseudo header. Wireless LAN: Transmission Medium For WLANs, MAC problems, Hidden and Exposed terminals, Near and Far terminals, Infrastructure and Ad hoc Networks, IEEE 802.11- System arch, Protocol arch, Physical layer, Concept of spread spectrum, MAC and its management, Power management.

UNIT 5 Mobile IP: unsuitability of Traditional IP; Goals, Terminology, Agent advertisement and discovery, Registration, Tunneling techniques. Ad hoc network routing: Ad hoc Network routing v/s Traditional IP routing, types of routing protocols, Examples: OADV, DSDV, DSR, ZRP. Mobile Transport Layer: unsuitability of Traditional TCP; I-TCP, S-TCP, MTCP.

DIRECT ASSESMENT:
ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Tanenbaum A. S., “Computer Networks”, Pearson Education, 5th edition, 2011.
2. Comer, “Internetworking with TCP/ IP Vol-1”, Pearson education, 6th Edition, 2015.
3. Jochen Schiller “Mobile communication”, 2nd edition, Pearson education, 2008

REFERENCE BOOKS:

1. W. Richard Stevens, “TCP/IP Illustrated Vol-1 ”, 2nd Edition, Addison-Wesley, 2011.

RESEARCH JOURNALS:

1. ACM/IEEE Symposium on Architectures for Networking and Communications Systems
2. ACM Transactions on Computer Systems

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A
CO 71018: ADVANCES IN OPERATING SYSTEMS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Fundamentals of Operating Systems, Programming Skills

COURSE OBJECTIVES: To enable a student in high level understanding of advanced operating systems and their applications in diverse domains.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Demonstrate understanding of design issues of advanced operating systems and Compare different types of operating systems.
2. Analyze design aspects for different subsystems of diverse Operating Systems.
3. Compare & Contrast different architectures used in Distributed Operating Systems
4. Analyze synchronization amongst various components of a typical operating system.

COURSE CONTENTS:
THEORY:

UNIT 1 Review of Operating System Fundamentals; Different Types, Dedicated Operating Systems; The Genesis of Modern Operating Systems; Operating Systems Design Strategies/Kernel Architectures – Microkernels, Exokernels etc.
Operating Systems Services, System Calls and their Implementation.

UNIT 2 File Systems and Main Memory Management: File Concept, Different Modules of a File System; File Protection; Disk Partitioning; Kernel I/O Subsystem; Advancements for improving File System Performance; System Calls for File Systems Management. Review of Main Memory Management Techniques including Virtual Memory.

UNIT 3 Process Management: Review of Process and Scheduling Concepts; System Calls for Process Management; IPC; Concept of Threads: Process v/s Threads, User Level & Kernel Level Threads, Threads Scheduling, Threading Issues, Solutions to Critical Section Problem and Synchronization for Threads;
Scheduling in Multi-core Systems. Load Balancing Techniques for Multicore and Multiprocessor systems- Dynamic Load Balancing, Process Migration.

UNIT 4 Distributed Operating Systems: Design Issues; Overview of Distributed File Systems, Distributed Process Management and Distributed Memory Management.
Embedded and IoT Operating Systems: Introduction, Characteristics and Features, Challenges and Issues in Designing the Operating Systems for Resource Constrained Systems.

UNIT 5 Virtualization: Basic Concepts, Benefits and Features, Building Blocks, Virtualization and O/S Components, Hypervisors; Virtual Machines, CPU and Memory Virtualization.
Case Studies: Unix/Linux, Windows and Contemporary Embedded Operating Systems like Embedded Linux; Study of Source Code of Open Source Operating System like Linux.

DIRECT ASSESMENT:
ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Silberschatz, Galvin, Gagne, "Operating System Concepts'', Wiley, 9/E.
2. William Stallings, "Operating Systems: Internals and Design Principles", Pearson.
3. Pradeep K. Sinha, "Distributed Operating System: Concept and Design'', PHI.
4. Matthew Portnoy, "Virtualization Essentials", Sybex.

REFERENCE BOOKS:

1. Andrew S. Tanenbaum, Albert S. Woodhull, "Operating Systems: Design and Implementation", Pearson Education.
2. Wang K.C., "Embedded and Real-Time Operating Systems", Springer.
3. Bovet &Cesati, "Understanding the Linux Kernel", O'Reily.
4. Maurice J. Bach, "The Design of Unix Operating System", Prentice Hall.

REFERENCE LINKS:-

1. http://www.kernel.org/
2. http://www.linux.org/
3. http://www.linuxquestions.org/

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-I)
CO 71221: MACHINE LEARNING

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Knowledge about Linear Algebra, Calculus, Python/R.

COURSE OBJECTIVES:

1. To understand various methods and algorithms in machine learning.
2. To analyze different hyperparameters of machine learning algorithms.
3. To analyze the real-world problem and find the feasibility of applying a machine learning algorithm to it.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define in-depth about theories, methods, and algorithms in machine learning.
2. Find and analyze the optimal hyper parameters of the machine learning algorithms.
3. Examine the nature of a problem at hand and determine whether machine learning can solve it efficiently enough.
4. Solve and implement the real world problems using machine learning.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to machine learning (ML): Basics of ML, History of ML, Evolution of ML, ML Models, Learning and testing models, ML Algorithm and Convergence, ML Techniques, Types of ML, supervised and unsupervised learning, classification and clustering, Applications of ML, Bias-Variance tradeoff, ML Tools.

UNIT 2 Neural Networks: McCulloch Pitts Neuron models, Activation Functions, Loss Functions, perceptron, Gradient Descent, Multilayer neural networks: back-propagation, back-propagation calculus, Initialization, Training rules, issues in back-propagation, Bayesian Learning, Competitive learning and self-organization map.

UNIT 3 Support Vector Machines(SVM): SVM Formulation, Interpretation & Analysis, hard and soft margin, Hinge loss, SVM dual, SVM tuning parameters, SVM Kernels, twin SVM.

UNIT 4 Clustering: K-Means Clustering, Mean Shift Clustering, Agglomerative clustering, Association Rule Mining, Partition Clustering, Hierarchical Clustering, Birch Algorithm, CURE Algorithm, Density-based Clustering, Gaussian Mixture Models, and Expectation Maximization. Parameters estimations – MLE, MAP

UNIT 5 Learning Theory: Probably Approximately Correct (PAC) Model, PAC Learnability, Agnostic PAC Learning, Theoretical analysis of machine learning problems and algorithms, Generalization error bounds, VC Model.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.
2. Leonard Kaufman and P. J. Rousseau. Finding groups in data: An introduction to cluster analysis, Wiley, 2005.
3. NelloCristianiniandJohn Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, 2000.
4. Bernhard SchölkopfandAlexander J. Smola, Learning with Kernels, MIT Press, 2002.
5. Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press,2014.

REFERENCE BOOKS:

1. Ian Goodfellow and YoshuaBengio and Aaron Courville, Deep Learning, An MIT Press book, 2016
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2010
3. Tan, Steinbach and Vipin Kumar, Introduction to Data Mining, Pearson Education, 2016.

RESEARCH JOURNALS:

1. IEEE Transaction on Neural Networks and Learning Systems.
2. ACM Journal of Machine Learning Research.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-I)
CO 71222: INFORMATION SECURITY

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basic Engineering Mathematics, Basic knowledge of Computer Networks

COURSE OBJECTIVES:
After completing the course student should be able to:

1. Understand importance of information security and know various security threats and mechanisms.
2. Apply cryptographic methods for securing information.
3. Design secure programs and find vulnerability in code.
4. Analyze security of a system.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define and explain different terms and existing security threats and vulnerabilities in information systems.
2. Design a secure computing system by applying suitable cryptographic mechanisms.
3. Assess security requirements in an information system and create suitable model.
4. Evaluate and analyze security in Computer Network and Software.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to Information Security, Security threats – Vulnerabilities and attacks, Security Goals, Legal and Ethical Issues in Computer Security.

UNIT 2 Cryptography – Classical Cryptography, Symmetric key Encryption: DES, Triple DES algorithm, Key distribution; Diffie-Hellman Key exchange; Public Key Cryptography: RSA algorithm, ECC, El-Gamal Algorithm; Hash Functions and Message Authentication: MD5, SHA-1, HMAC, Applications of cryptography

UNIT 3 PKI: Digital Signatures, Digital Certificates, X.509 standard, Authentication applications like Kerberos/Needham-Schroder algorithm; Access Control –Discretionary, Mandatory, Role Based Mechanisms, Security Policies: Definition, Types, various models of security

UNIT 4 Program Security: Unexpected Behaviour, Types of Flaws, Program Errors: Buffer Overflows, Incomplete Mediation, Time-of-Check to Time-of-Use Errors; Covert channels, Controls against program threat; Viruses and other malicious code, types of malicious codes; symptoms of infection, How malicious code works. User Authentication mechanisms

UNIT 5 Network Security Mechanisms- Vulnerabilities in Computer Network protocols, Threats to computer network, Security Protocols like IPSec, SSL, TLS, SET; Firewalls- types of firewall, DMZ, Intruders, Intrusion detection and prevention techniques

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. W. Stallings, "Cryptography and Network Security – Principles and Practices", Latest Edition, Pearson Education,
2. Pfleeger and Pfleeger, "Security in Computing", Latest Edition, Pearson Education
3. Whitman and Mattord, "Principles of Information Security", Thomson Course Technology, Latest Edition, Vikas Publishing House

REFERENCE BOOKS:

1. Bernard Menezes, Network Security and Cryptography, Latest edition, Cengage Learning,
2. Bishop, Venkataramanayya, "Introduction to Computer Security", Pearson Education.
3. Mann, Mitchell, Krell, "Linux System Security", Latest Edition, Pearson Education
4. Robert, C. Newman, "Enterprise Security", Latest Edition, Pearson Education.
5. Kaufman, Perlman and Speciner, "Network Security, Private Communication in a Public Network", Latest Edition, Prentice Hall of India.
6. Nortcutt& Judy Novak, "Network Intrusion Detection", latest Edition, Pearson Education.

MOOCS COURSES:

1. Introduction to Information Security-1,2,3,4,5, Prof. V. Kamakoti.
2. Cryptography and Network Security, Prof.SauravMukhopadhyay.

RESEARCH JOURNALS:

1. ACM Journal on Information and system security.
2. IEEE Journal on Dependable and Secure Computing

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-I)
CO 71224: AUGMENTED REALITY AND VIRTUAL REALITY

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Computer Graphics.

COURSE OBJECTIVES:
This course provides students with an opportunity to explore the research issues in Augmented Reality and Virtual Reality (AR & VR). It also makes the students know the basic concept and framework of virtual reality.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Explain the fundamental concepts of virtual reality and models of input and output interface in virtual reality.
2. Compare and Contrast between different development tools and frameworks used in virtual reality.
3. Design and apply the concepts of virtual reality in various applications.
4. Articulate the basic concepts of augmented reality and its various applications.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction of Virtual Reality: Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality. Multiple Models of Input and Output Interface in Virtual Reality: Input -- Tracker, Sensor, Digital Glove, Movement Capture, Video-based Input, 3DMenus & 3DScanner etc. Output -- Visual/Auditory/Haptic Devices.

UNIT 2 Visual Computation in Virtual Reality: Fundamentals of Computer Graphics. Software and Hardware Technology on Stereoscopic Display. Advanced Techniques in CG: Management of Large Scale Environments & Real Time Rendering.

UNIT 3 Interactive Techniques in Virtual Reality: Body Track, Hand Gesture, 3D Manus, Object Grasp. Development Tools and Frameworks in Virtual Reality: Frameworks of Software Development Tools in VR. X3D Standard; Vega, MultiGen, Virtools etc.

UNIT 4 Application of VR in Digital Entertainment: VR Technology in Film & TV Production. VR Technology in Physical Exercises and Games. Demonstration of Digital Entertainment by VR.

UNIT 5 Augmented and Mixed Reality: Taxonomy, technology and features of augmented reality,

difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, enhancing interactivity in AR environments, evaluating AR systems.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Burdea, G. C. and P. Coffet. Virtual Reality Technology, Second Edition. Wiley-IEEE Press, 2003/2006.
2. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013.
3. D. Schmalstieg and T. Höllerer. Augmented Reality: Principles and Practice. Addison-Wesley, Boston, 2016.

REFERENCE BOOKS:

1. Tony Parisi, Learning Virtual Reality, 2015, O'Reilly Publishers.
2. Jorge R. Lopez Benito, EnaraArtetxe Gonzalez, Enterprise Augmented Reality Projects, 2019, Packt Publishers.
3. Erin Pangilinan, Steve Lukas, Vasanth Mohan, Creating Augmented and Virtual Realities: Theory and Practice for Next-Generation Spatial Computing, 2019, O'Reilly Publishers.

RESEARCH JOURNALS:

1. IEEE Transactions on Visualization and Computer Graphics.
2. ACM Transactions on Multimedia Computing, Communications, and Applications.
3. ACM Transactions on Graphics.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-I)
CO 71225: MULTIMEDIA SYSTEM

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basics of Computer Graphics.

COURSE OBJECTIVES: To enable a student in high level understanding of multimedia systems, their technologies, standards and their applications in diverse domains.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Describe the relevance and underlying infrastructure of the multimedia systems.
2. Define core multimedia technologies and standards (Digital Audio, Graphics, Video, VR, data transmission/compression).
3. Analyze the factors involved in multimedia systems performance, integration and evaluation.
4. Design and implement a multimedia system.

COURSE CONTENTS:
THEORY:

UNIT 1 **Introduction to Multimedia System:** Architecture and components, Multimedia distributed processing model, Synchronization, Orchestration and Quality of Service (QOS) architecture.

UNIT 2 **Audio and Speech, Images and Video:** Data acquisition, Sampling and Quantization, Human Speech production mechanism, Digital model of speech production, Analysis and synthesis, Psycho-acoustics, low bit rate speech compression, MPEG audio compression. Image acquisition and representation, Bilevel image compression standards: ITU Group III and IV standards, JPEG image compression standards, MPEG video compression standards.

UNIT 3 **Multimedia Communication:** Fundamentals of data communication and networking, Bandwidth requirements of different media, Real time constraints: Audio latency, Video data rate, multimedia over wired and wireless networks, Multimedia conferencing.

UNIT 4 **Hypermedia presentation:** Authoring and Publishing, Linear and non-linear presentation, Structuring Information, Different approaches of authoring hypermedia documents, Hyper-media data models and standards.

UNIT 5 **Multimedia Information Systems:** Operating system support for continuous media applications: limitations is usual OS, New OS support, Media stream protocol, file system support for continuous media, multimedia storage data models for multimedia and hypermedia information, content based retrieval of unstructured data.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Ralf Steinmetz and Klara Nahrstedt, Multimedia Systems, Springer.
2. J. D. Gibson, Multimedia Communications: Directions and Innovations, Springer.
3. K. Sayood, Introduction to Data Compression, Morgan-Kaufmann.
4. A.Puri and T. Chen, Multimedia Systems, Standards, and Networks, Marcel Dekker.
5. BorivojeFurht, Handbook of Multimedia Computing, CRC Press.

REFERENCE BOOKS:

1. Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, John Wiley.
2. Colton McAnlis, AleksHaecky, Understanding Compression, 2016, O'Reilly Media, Inc.
3. Jitae Shin, Daniel C. Lee, C.-C. Jay Kuo, Quality of Service for Internet Multimedia, 2003, Pearson Publications.
4. Vic Costello, Multimedia Foundations, 2nd Edition, 2016, Routledge Publishers.
5. Ling Guan, Multimedia Image and Video Processing, 2nd Edition, 2017, CRC Press.
6. Benny Bing, Next-Generation Video Coding and Streaming, 2015, Wiley Publications.

RESEARCH JOURNALS:

1. IEEE Transactions on Multimedia
2. ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP)

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-II)
CO 71311: COMPUTATIONAL INTELLIGENCE

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Computer Programming, Knowledge of Machine Learning Tools.

COURSE OBJECTIVES:

1. To introduce basic concepts, theories and techniques of computational intelligence.
2. Help students to learn the applications of computational intelligence techniques in the diverse fields of science, engineering, medicine, finance etc.

COURSE OUTCOMES:

After completing the course student should be able to:

1. Describe in-depth about theories, methods, and algorithms in computational Intelligence.
2. Compare and contrast traditional algorithms with nature inspired algorithms.
3. Examine the nature of a problem at hand and determine whether a computation intelligent technique/algorithm can solve it efficiently enough.
4. Design and implement Computation Intelligence algorithms and approaches for solving real-life problems.

COURSE CONTENTS:

THEORY:

UNIT 1 Introduction to Computational Intelligence (CI): Basics of CI, History of CI, Adaptation, Learning, SelfOrganization and Evolution, CI and Soft Computing, CI Techniques; Applications of CI; Decision Trees: Introduction, Training Decision Trees, Evaluation, Splitting Criteria, Decision Tree Induction Algorithms.

UNIT 2 Evolutionary Computation: Genetic Algorithms: Basic Genetics, Concepts, Working Principle, Creation of Offsprings, Encoding, Fitness Function, Selection Functions, Genetic Operators-Reproduction, Crossover, Mutation; Genetic Modeling, Benefits; Problem Solving; Introduction to Genetic Programming, Evolutionary Programming, and Evolutionary Strategies.

UNIT 3 Fuzzy System: Fuzzy Sets: Formal Definitions, Membership Functions, Fuzzy Operators, Fuzzy Set Characteristics, Fuzzy Relations and Composition, Fuzziness and Probability; Fuzzy Logic and Reasoning: Fuzzy Logic, Fuzzy Rules and Inferencing; Fuzzy Controllers: Components of Fuzzy Controllers, Types, Defuzzification.

UNIT 4 Rough Set Theory: Introduction, Fundamental Concepts, Knowledge Representation, Set Approximations and Accuracy, Vagueness and Uncertainty in Rough Sets, Rough Membership Function, Attributes Dependency and Reduction, Application Domain, Hidden Markov Model (HMM),Graphical Models, Variable Elimination, Belief Propagation,Markov Decision Processes.

UNIT 5 Swarm Intelligence: Introduction to Swarm Intelligence, Swarm Intelligence Techniques: Ant Colony Optimization (ACO): Overview, ACO Algorithm; Particle Swarm Optimization (PSO): Basics, Social Network Structures, PSO Parameters and Algorithm; Application Domain of ACO and PSO; Bee Colony Optimization, Grey wolf optimization etc.; Hybrid CI Techniques and applications; CI Tools.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Russell C. Eberhart and Yuhui Shi, Computational Intelligence: Concepts to Implementations, Morgan Kaufmann Publishers, 2007.
2. Andries P. Engelbrecht, Computational Intelligence: An Introduction, Wiley Publishing, 2007.
3. David E. Goldberg, Genetic Algorithm in Search Optimization and Machine Learning, Pearson Education, 2009.
4. Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Springer, 1991.

REFERENCE BOOKS:

1. Jagdish Chand Bansal, Pramod Kumar Singh, Nikhil R. Pal, Evolutionary and Swarm Intelligence Algorithms, Springer Publishing, 2019.
2. S. Rajasekaran, G.A. VijaylakshmiPai, “Neural Networks, Fuzzy Logic, Genetic Algorithms Synthesis and Applications”, PHI, 2003.

RESEARCH JOURNALS:

1. IEEE Transactions on Evolutionary Computation
2. IEEE Transactions on Systems, Man and Cybernetics
3. IEEE Transaction on Neural Networks and Learning Systems
4. IEEE Transaction on Fuzzy Systems
5. IEEE Transactions on Pattern Analysis and Machine Intelligence
6. ACM Transactions on Intelligent Systems and Technology
7. ACM Genetic and Evolutionary Computation Conference (GECCO)
8. ACM Journal of Machine Learning Research

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-II)
CO 71312: CLOUD COMPUTING

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Computer Network.

COURSE OBJECTIVES: Students will be able to understand the concept of service oriented architecture and gain knowledge of various cloud servicemodel, deployment model, type of virtualization. Get detailed knowledge about cloud resource management, and security issues.

COURSE OUTCOMES:

1. Explain fundamental cloud computing and its different types service models and deployment models.
2. Explain importance of virtualization and differentiate various types of hypervisors.
3. Compare and contrast different cloud file systems and demonstrate working of map reduce.
4. Evaluate the performance of cloud computing and Analyze security and resource optimization issues related to cloud computing environment.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to Service Oriented Architecture, RESTful services: Introduction to cloud computing: Definition, Characteristics, cloud services, Types; Software as a Service, Platform as a Service, Infrastructure as Service, Deployment model, Architecture of cloud computing, Application, benefits and limitations, basic technologies of cloud computing, Multi-tenancy and scalabilty.

UNIT 2 Concept of Virtualization, characteristics, Virtualization applications in enterprises, Pitfalls of virtualization, reference model of virtualization, Hypervisor, types of hypervisor, types of virtualization, case study of hypervisors.

UNIT 3 Data in the cloud: Relational databases, Cloud file systems: GFS and HDFS, Features and comparisons among GFS, HDFS etc, BigTable, HBase and Dynamo. Map-Reduce and extensions: Parallel computing, The Map-Reduce model: Parallel efficiency of Map-Reduce, Relational operations, Enterprise batch processing, Example/Application of Map-Reduce.

UNIT 4 Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud: Cloud computing security architecture, General Issues, Trusted Cloud computing, Security challenges: Virtualization security management-virtual threats, VM Security Recommendations, VM-Specific Security techniques, Secure Execution Environments and Communications in cloud.

UNIT 5 Issues in cloud computing; implementing real time application; QOS Issues in Cloud, Dependability, data migration, streaming in Cloud. Cloud Middleware. Mobile Cloud Computing. Inter Cloud issues. A grid of clouds, Sky computing, load balancing, Resource optimization, Resource dynamic reconfiguration, Monitoring in Cloud, Installing cloud platforms and performance evaluation, Features and functions of cloud computing platforms.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Kai Hawang, Geoferrey C Fox, "Distributed and Cloud Computing", Elseveir publication,2012.
2. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley India Edition.
3. RajkumarBuyya, Christian Vecchiola, S. Thamaraselvi, Mastering Cloud Computing, McGraw Hill, 2013.

REFERENCE BOOKS:

1. Scott Granneman, "Google Apps", Pearson, 2012.
2. Tim Malhar, S.Kumaraswammy, S.Latif, "Cloud Security & Privacy", SPD, O'REILLY.
3. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India, 2011.

RESEARCH JOURNALS:

1. Scott Granneman, "Google Apps", Pearson, 2012.
2. Tim Malhar, S. Kumaraswammy, S.Latif, "Cloud Security & Privacy", SPD, O'REILLY.
3. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India, 2011.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-II)
CO 71313: OBJECT ORIENTED SOFTWARE ENGINEERING

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Object-Oriented and Software Engineering Concepts.

COURSE OBJECTIVES:
To expose students to formal processes for the design, implementation and management of large software systems using object-oriented software development approach.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Compare and Contrast Object-Oriented Software Development and Conventional Software Development.
2. Model different perspectives of Object-Oriented Software Analysis and Design using UML.
3. Apply Object-Oriented design, development and testing principles in the construction of real world software systems of varying complexities.
4. Apply quality standards for object oriented systems.

COURSE CONTENTS:
THEORY:

UNIT 1 **Review of Software Engineering and Object Orientation:** The Object Oriented Paradigm, Object Oriented Principles, Need for Object Oriented Approach, Problem Solving using Object Oriented approach; Object Oriented Software Development Models, Rational Unified Process:Basic Concepts, Symptoms and Root Causes of Software Development Problems, Best Practices of RUP, RUP software life cycle, 4+1 view model, Various Workflows.

UNIT 2 **Object-Oriented Modeling with UML:** Object-Oriented Modeling Concepts, Object Modeling Technique(OMT),Unified Modeling Language (UML) basic Notations, Extensibility Mechanisms, Multiplicity, Common Modeling Techniques for modeling Relationships: Realization, Dependencies, Generalization, Associations: Aggregation and Composition; Visual Modeling, Guidelines for structural and behavioral Modeling, Study of UML tools.

UNIT 3 **Object Oriented Analysis:** Object Oriented Techniques for classes and object identification, Techniques for gathering requirements, Requirement analysis, Requirements Review, Managing changing requirements, Use case driven Approach, Business Modeling using activity diagrams, Domain Modeling using class diagrams,CRC cards design.

UNIT 4 **Object Oriented Design and Development:** Conventional v/s OO design, Design Principles, Design Guidelines: apply classes and Objects concept in real world

applications, User Interface Design, Modeling interactions and behavior: Interaction Diagrams, state diagram, Difficulties and risks; Mapping model to code, Component Based Development: Designing of Components and interfaces with Component and deployment Diagram.

UNIT 5 **Object Oriented Testing:** Correctness and Consistency of OOA & OOD models,Impact of object orientation on Testing, Types of testing, Testing Strategies, test cases and test plans for OO software process, Object oriented Project Management, Software Quality Assurance, Software Configuration Management, Documentation Standards, Case studies like ATM, Virtual Classroom System etc.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Roger Y. Lee, "Object-Oriented Software Engineering with UML: A Hands-On Approach", Nova Science Publishers Inc, 15 Feb 2019.
2. Ivar Jacobson, "Object Oriented Software Engineering", Pearson Education, 2nd Edition 2011.
3. Stephen R. Schach, "Object Oriented Classical Software Engg." Tata McGraw Hill, 8thedition, Dec 2010.
4. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modelling Language User Guide", Pearson Education 2008.
5. Gamma G.Helm, Johnson, "Design Patterns, Elements of Reusable Object Oriented Software", Addison Wesley. 1st Edition.

REFERENCE BOOKS:

1. Ivon Jacobson, "Object Oriented Software Engineering", Addison Wesley. Booch G., "The Unified Modelling User Guide".
2. Timothy C. Lethbridge, Robert Laganiere, "Object Oriented Software Engg.", Tata McGraw Hill, 2004.
3. Phillipe Kruchten, "The Rational Unified Process - An Introduction", Pearson Ed. 2000.
4. Ivar J, Grady B, James R., "The Unified Software Development Process", Pearson Ed. 2003.
5. IBM Rational Modules.

RESEARCH JOURNALS:

1. IEEE Transactions on Software Engineering
2. IEEE Transactions on Object Oriented Programming
3. ACM Transaction on Programming Language and Systems

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-II)
CO 71314: ROBOTICS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basic of Mechanics.

COURSE OBJECTIVES:
To provide the basic understanding to the students related to the robots, various types of robots and their working.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define and identify the basic components of robotic systems.
2. Describe different types of kinematics and dynamics of robots.
3. Design and execute the basic program and functions for robotics.
4. Design and develop the basic robots for specific tasks.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction - History of robots, Classification of robots, Present status and future trends. Basic components of robotic systems. Basic terminology- Accuracy, Repeatability, Resolution, Degree of freedom. Mechanisms and transmission, End effectors, Grippers-different methods of gripping, Mechanical grippers-Slider crank mechanism, Screw type, Rotary actuators, Cam type gripper, Magnetic grippers, Vacuum grippers, Air operated grippers; Specifications of robot.

UNIT 2 Drive systems and Sensors- Drive system- hydraulic, pneumatic and electric systems Sensors in robot – Touch sensors, Tactile sensor, Proximity and range sensors, Robotic vision sensor, Force sensor, Light sensors, Pressure sensors.

UNIT 3 Kinematics and Dynamics of Robots 2D, 3D Transformation, Scaling, Rotation, Translation, Homogeneous coordinates, multiple transformations, Simple problems. Matrix representation, Forward and Reverse Kinematics Of Three Degree of Freedom, Homogeneous Transformations, Inverse kinematics of Robot, Robot Arm dynamics, D-H representation of robots, Basics of Trajectory Planning.

UNIT 4 Robot Control, Programming and Applications Robot controls-Point to point control, Continuous path control, Intelligent robot, Control system for robot joint, Control actions, Feedback devices, Encoder, Resolver, LVDT, Motion Interpolations, Adaptive control.

UNIT 5 Introduction to Robotic Programming, On-line and off-line programming, programming examples. Robot applications and case studies.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Mikell P Groover, Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, “Industrial Robotics, Technology programming and Applications", McGraw Hill, 2012.
2. Craig. J. J. “Introduction to Robotics- mechanics and control”, Addison- Wesley, 1999.

REFERENCE BOOKS:

1. Richard D. Klafter, Thomas. A, ChriElewski, Michael Negin, "Robotics Engineering an Integrated Approach", PHI Learning., 2009.
2. Francis N. Nagy, AndrasSiegler, "Engineering foundation of Robotics", Prentice Hall Inc., 1987.
3. Carl D. Crane and Joseph Duffy, "Kinematic Analysis of Robot Manipulators", Cambridge University press, 2008.
4. Fu. K. S., Gonzalez. R. C. & Lee C.S.G., “Robotics control, sensing, vision and intelligence”, McGraw Hill Book co, 1987.

RESEARCH JOURNALS:

1. IEEE Transactions on Robotics
2. ACM Transactions on Human-Robot Interaction

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-A (ELECTIVE-II)
CO 71315: GAME DESIGN

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Programming Concepts (C/C++/Java/Python).

COURSE OBJECTIVES:

1. Understand the core concepts of game design.
2. Explore the structure, methods, and economics of the games industry.
3. Learn level and character design and development.
4. Have end to end gaming experience.

COURSE OUTCOMES:

After completing the course student should be able to:

1. Demonstrate fundamental game design principles.
2. Define game development process.
3. Design and build characters and game environments.
4. Design and develop a complete game as per specifications.

COURSE CONTENTS:

THEORY:

UNIT 1 Brief History of Game Platforms, Modern Game Platforms, Project Overview, Getting Started
Creating Tree Sprite, Creating Ground Tiles, Creating Platforms, Setting up the Player, Tags, Layers
Sorting Layers, Player Animations, Player Animator Controller, Character Controller, Using Game Controllers.

UNIT 2 Gameplay Design, Mechanics & Dynamics, Level Design, Level Components, Level Designing practices, Game Balance, Mechanic Balance, Moving Platforms, Project Organization, Camera Follow

UNIT 3 Character Controller Code, Defining behavior of characters, like adding Double Jump, Death Zone, Coins, Victory, Particle Effects Enemy Setup Enemy Animation, Attacking and Stunning Enemy, Moving Enemy, Enemy Code, Player Bounce on Enemy Stun.

UNIT 4 Building Out the Game: Parallax Scrolling Part , Image Effects, User Interface Setup, Hiding UI in the Scene View, Game Manager, Player Preferences, Game Manager Code, Custom Editor Tools, Out-of-Game Scenes, The Main Menu, Cross Platform Menu Controller Support, Supporting Mobile Cross Platform Input.

UNIT 5 Preparing to Ship: Building for WebGL, Building for WebGL, Building for PC and Mac, Building for iOS1, Building for Android, Taking Screenshots, Creating a Game Web Page, game design economics.

DIRECT ASSESMENT:
ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Jesse Schell, The Art of Game Design, 3rd Edition, 2019, CRC Press.
2. Nystrom Robert,Game Programming Patterns, 3rd edition, Genever Benning, 2014.

REFERENCE BOOKS:

1. Anna Anthropy and Naomi Clark, A Game Design Vocabulary: Exploring theFoundational Principles Behind Good Game Design, 2014, Addison-Wesley Professional.
2. Sung, K., Pavleas, J., Arnez, F., Pace, J., Build Your Own 2D Game Engine and Create Great Web Games: Using HTML5, JavaScript, and WebGL, 2015, Apress Publishers.
3. Zach Hiwiller, Players Making Decisions, 2nd Edition, 2019, New Riders Publishers.
4. Patrick Alessi, Beginning IOS Game Development, 2011, Wrox Publishers.

RESEARCH JOURNALS:

1. IEEE Transactions on Games
2. IEEE Transactions on Computational Intelligence and AI in Games

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B
CO 71511: DATABASE ENGINEERING

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Fundamentals ofOperating Systems and DBMS.

COURSE OBJECTIVES:
To provide students with basic concepts in databases both in terms of usage and implementation.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Compare and contrast DBMS with traditional file systems and be able to use concepts of Entity-Relationship model to design a database for given domain and apply normalization techniques for a good design
2. Illustrate the knowledge of SQL for retrieving data from database, and apply, analyze and plan the process of query optimization.
3. Explain and apply the basic concepts of transaction processing, concurrency control and solve the problems related to storage structures and access techniques.
4. Illustrate use of distributed and parallel databases in applications.

COURSE CONTENTS:
THEORY:

UNIT 1 Review of Basic Concepts of DBMS: Database system architecture, Conceptual modeling of database, Normalization theory, Query languages: Relational Algebra, SQL – Joins, nested queries, DDL, DML constraints, PL/SQL, Indexing, and Dynamic SQL.

UNIT 2 Query Optimization: Introduction, Measures of Query Cost, Various algorithms to implement select, project & join operation of relational algebra, Sorting, Selectivity Estimation, Improving query performance with variant indices.

UNIT 3 Transaction Processing: Transaction Concepts, Isolation, Concurrent Executions, Serializability, Recoverability. Concurrency Control Protocols, Deadlock: Prevention and Recovery, Various Recovery methods with Concurrent Transactions, Buffer Management, and Failure with Loss of Nonvolatile Storage.

UNIT 4 Distributed Database System: Distributed database architecture, Heterogeneous and Federated Database Systems, Distributed database design, Distributed Query processing, Distributed Transaction processing, Fragmentation and Distribution Parallel Database Systems, Deductive database system, Object Relational Database Systems.

UNIT 5 Advance Topics: Brief review of Data mining, Data Ware Housing, Client server computing, XML database and web mining.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Korth H.F. &Silberschatz A., Sudarshan, “Database Systems”, McGraw-Hill, Seventh edition, 2019.
2. Elmasri R., Navathe S.B., “Fundamentals of Database Systems”, The Benjamin/ Cummings Publishing Company, Pearson. seventh edition Inc., 2015.
3. George Coulouris, “Distributed systems concepts and Design”, Pearson Education, fifth edition, 2017.
4. Tannenbaun and Van Steen, “Distributed systems principles and paradigm”. PHI, 2007.

REFERENCE BOOKS:

1. D.Ullman, J.Widom, “Database Systems: The Complete Book”, Pearson Education, 2011.
2. Alexis Leon, Mathews Leon, “Database Management Systems”, Vikas Publishing House PvtLtd , New Delhi, 2008.

WEB SITES:

1. www.cs.cornell.edu/courses/cs632/2001 sp
2. www.ieee.org

RESEARCH JOURNALS:

1. ACM Transaction on Database Systems.
2. The International Journal on Very Large Data Bases

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B
CO 71512: ALGORITHMICS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Data Structures, Computer Programming.

COURSE OBJECTIVES:

1. Analyze the asymptotic performance of algorithms.
2. Visualize usage of major algorithms and data structures.
3. Apply different algorithm design and analysis methods to common engineering design situations.

COURSE OUTCOMES:

After completing the course student should be able to:

1. Analyze the asymptotic performance of algorithms.
2. Design algorithms for various problems using different design paradigms
3. Compare different data structures and employ these to model engineering problems where appropriate.
4. Define computability, non-computability and various complexity classes.

COURSE CONTENTS:

THEORY:

UNIT 1 Review study: ADT, RAM Model, Characteristics and Properties of algorithms. Algorithm Design, Analysis of Algorithm, Time and Space Complexity issues, Worst case and Average case analysis. Asymptotic Notations: O, o, Ω, ω, θ, Solving Recurrences-Recursion tree method, Master Method; Amortized Analysis.

UNIT 2 Advanced Data Structure Topics: Binary Search Tree, AVL Tree, B-Tree, Red-Black Tree, Hashing- Hash tables, Hash functions, Collision handling methods. Heaps: Binary, Binomial, Fibonacci; Heap Sort.

UNIT 3 Divide and conquer: Quick sort, Merge Sort etc.; Finding median, Counting Inversion, Linear time Sorting; Binary search, Dynamic Programming: Concept, Dynamic programming Vs Divide and Conquer, Matrix Chain multiplication, 0/1 knapsack problem.

UNIT 4 Greedy Algorithm: Basic Concept, fractional knapsack problem, Scheduling problem. Graph algorithms: DFS, BFS, Topological sorting, shortest path, Minimum Spanning Tree, Network flow Problem.

UNIT 5 Solvable and Unsolvable problems: P, NP, NP Hard and NP complete problems, Cooks theorem, reduction, NP-Complete problems: Satisfiability, Clique problem, Independent Set problem (ISP), Introduction to Approximation Algorithms: TSP. Introduction to Parallel Algorithms, PRAM model, Threaded Fibonacci algorithm. Number Theoretic algorithms.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. T.H. Cormen, C.E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms", PHI-EEE, 2003.
2. Jon Kleinberg, Eva Tordos, "Algorithm Design", Pearson Edu, 2006.
3. Ellis Horowitz, SartajSahani, SanguthevarRajsekaran," Fundamentals of Computer Algorithms", 2nd Edition, University Press, 2010.

REFERENCE BOOKS:

1. Aho, Hopcroft, Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education, Asia, 2002.
2. G. Brassard and P. Bratley, "Fundamentals of Algorithms", Prentice Hall of India 2003.
3. Michael T. Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples", John Wiley& Sons Inc., 2002.
4. D.E. Knuth, "The Art of Computer Programming", Vol.1, 2, 3, Second Edition, Pearson Education, 2004.

WEB SITES:

1. www.algorithm.net
2. www.njit.edu
3. www.iitb.ac.in

RESEARCH JOURNALS:

1. ACM Journal on Algorithms and Computation Theory.
2. ACM Journal of Experimental Algorithms.
3. IEEE/ACM Journal on Algorithmics.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B
CO 71513: AGILE SOFTWARE DEVELOPMENT

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Fundamentals ofSoftware Engineering.

COURSE OBJECTIVES:

1. Understand the differences between conventional and agile approaches.
2. Learn the background and origins of various agile concepts and methodologies.
3. Understand frameworks and practices used by agile teams.
4. Learn about agile ways of gathering requirements, estimation, release planning, performance metrics, and scaling.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Compare and Contrast agile software development model with traditional development models.
2. Apply agile design principles and practices to create high-quality software.
3. Build agile teams and Projects for effective software development using Scrum Framework.
4. Apply various agile testing techniques for Test-Driven Development.

COURSE CONTENTS:
THEORY:

UNIT 1 **Review of Software Engineering:** Software Engineering Concepts and Process; Software Development Life Cycle: Important Steps and Effort Distribution; Overview of Software Development Models like Prototype Model, Incremental Model, Spiral Model, RAD.

UNIT 2 **Fundamentals of Agile Development Methodology:** The Genesis of Agile, Introduction and background, Agile Manifesto and principles, Agile development Lifecycle, Agile Development Methods: Adaptive Software Development (ASD), Dynamic Systems Development Methods (DSDM), Scrum, Extreme Programming (XP): XP lifecycle, Feature Driven development, Lean Software Development, Kanban, Agile project management, Test Driven Development, Key Principles, Examples, Tools & Techniques for each Agile development methods.

UNIT 3 **Agile Requirements, Design and Development:** Impact of Agile Processes in Requirement Engineering, Requirements Elicitation and Management, Agility in Design, Agile Architecture, Agile Design Practices, Role of Design Principles, Agile Product Development, and Automated build tools, Continuous Integration, Continuous Deployment, Refactoring, Team Dynamics and Collaboration.

UNIT 4 **Agile Framework:** Introduction to Scrum, Agile Principles - Sprints Introduction, User Stories and Product Backlog, Estimation, Velocity, Burndown chart, Sprint Zero, Roles -

Team Management and Structures, Product Owner, ScrumMaster / Team Lead, Team Members, Planning in Scrum - Planning, Planning Stakeholders, Planning Types (Portfolio, Product and Sprint), Sprint phases/meeting - Sprint Planning, Sprint Review, Sprint Retrospective, Product Demo, Daily Scrum calls.

UNIT 5 **Agile Testing and Review:** Agile Testing Principles, Practice and Processes, Comparison between Testing in Traditional and Agile Approaches, Agile testing methods, techniques and tools, Estimating Test Efforts, Agile Metrics and Measurements, Agile Control: the 7 control parameters; Product Quality, Agile approach to Risk, Agile Approach to Configuration Management, Agility and Quality Assurance; Case study using any one of the frameworks.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Robert C. Martin, Agile Software Development- Principles, Patterns and Practices, Prentice Hall, 2013.
2. Kenneth S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison Wesley, 2012.
3. James Shore and Shane Warden, The Art of Agile Development, O'Reilly Media, 2007.
4. Craig Larman, ―Agile and Iterative Development: A manager's Guide, Addison-Wesley, 2004.
5. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson, 2001.

REFERENCE BOOKS:

1. Cohn, Mike, Agile Estimating and Planning, Pearson Education, 2006.
2. Cohn, Mike, User Stories Applied: For Agile Software Development Addison Wesley, 2004.
3. Agile Testing: A Practical Guide for Testers and Agile Teams by Lisa Crispin, Janet Gregory, Addison Wesley.
4. Agile Software Development: The Cooperative Game by Alistair Cockburn, Addison Wesley.

RESEARCH JOURNALS:

1. IEEE Transactions on Software Engineering
2. IEEE Transactions on Dependable and Secure Computing
3. IET Software
4. ACM Transactions on Software Engineering and Methodology (TOSEM)
5. ACM SIGSOFT Software Engineering Notes

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-III)
CO 71720: DEEP AND REINFORCEMENT LEARNING

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Machine Learning, Programming.

COURSE OBJECTIVES:
To provide a comprehensive skill for designing and implementation of real-life application development using the deep and reinforcement learning approaches.

COURSE OUTCOMES:
After completing the course student should be able to:
1. Define in-depth about theories, models and algorithms in machine learning.
2. Compare and contrast different learning algorithms with parameters.
3. Examine the nature of a problem at hand and find the appropriate learning algorithms and it's parameters that can solve it efficiently enough.
4. Design and implement of deep and reinforcement learning approaches for solving real-life problems.

COURSE CONTENTS:
THEORY:

UNIT 1 History of Deep Learning, McCulloch Pitts Neuron, Thresholding Logic, Activation functions, Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam, Eigenvalue Decomposition. Recurrent Neural Networks, Backpropagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, GRU, LSTMs, Encoder Decoder Models, Attention Mechanism, Attention over images.

UNIT 2 Autoencoders and relation to PCA, Regularization in autoencoders, Denoisingautoencoders, Sparse autoencoders, Contractive autoencoders, Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Batch Normalization, Instance Normalization, Group Normalization.

UNIT 3 Greedy Layerwise Pre-training, Better activation functions, Better weight initialization methods, Learning Vectorial Representations Of Words, Convolutional Neural Networks, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet, Visualizing Convolutional Neural Networks, Guided Backpropagation, Deep Dream, Deep Art, Recent Trends in Deep Learning Architectures.

UNIT 4 Introduction to reinforcement learning(RL), Bandit algorithms – UCB, PAC, Median Elimination, Policy Gradient, Full RL & MDPs, Bellman Optimality, Dynamic Programming - Value iteration, Policy iteration, and Q-learning & Temporal Difference Methods, Temporal-Difference Learning, Eligibility Traces, Function Approximation,

Least Squares Methods

UNIT 5 Fitted Q, Deep Q-Learning , Advanced Q-learning algorithms , Learning policies by imitating optimal controllers , DQN & Policy Gradient, Policy Gradient Algorithms for Full RL, Hierarchical RL,POMDPs, Actor-Critic Method, Inverse reinforcement learning, Maximum Entropy Deep Inverse Reinforcement Learning, Generative Adversarial Imitation Learning, Recent Trends in RL Architectures.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Deep Learning, An MIT Press book, Ian Goodfellow and YoshuaBengio and Aaron Courville.
2. Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition.
3. Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn van Otterlo, Eds.

REFERENCE BOOKS:

1. Pattern Classification- Richard O. Duda, Peter E. Hart, David G. Stork, John Wiley & Sons Inc.

RESEARCH JOURNALS

1. IEEE Transactions on Neural Networks and Learning Systems
2. Journal of Machine Learning Research

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-III)
CO 71721: MACHINE LEARNING FOR SECURITY

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basic course on Information Security**.**

COURSE OBJECTIVES:

1. To enable students to understand different machine learning techniques and their application in domain of security.
2. To understand data collection from a computer network and system for application of machine learning.

COURSE OUTCOMES:

After completing the course student should be able to:

1. Describe supervised and un-supervised machine learning models.
2. Apply machine learning techniques to different real-world information security problems.
3. Analyze data from various system and networks in order to perform analysis and make predictions.
4. Analyze the performance parameters of machine learning models in information security domain.

COURSE CONTENTS:

THEORY:

UNIT 1 **Classification and Clustering-**Machine Learning: Problems and Approaches, Model Families, Loss Functions, Optimization; Supervised Classification Algorithms- Logistic Regression. Decision Trees, Decision Forests, Support Vector Machines, Naive Bayes, k-Nearest neighbors, Neural Networks; Practical Considerations in Classification: Selecting a Model Family, Training data Construction, Feature Selection, Overfitting and Underfitting, Choosing Thresholds and Comparing Models, Clustering: Clustering Algorithms, Evaluating Clustering Results

UNIT 2 **Anomaly Detection-** Anomaly Detection Versus Supervised Learning, Intrusion Detection with Heuristics, Data-Driven Methods, Feature Engineering for Anomaly Detection: Host Intrusion Detection, Network Intrusion Detection, Web Application Intrusion Detection, Anomaly Detection with Data and Algorithms: Forecasting Supervised machine Learning, Statistical Metrics, Goodness-of-Fit, Unsupervised Machine Learning Algorithms, Density-Based Methods, Challenges of Using Machine Learning in Anomaly Detection, Response and Mitigation, Practical System Design Concerns, Optimizing for Explainability, Maintainability of Anomaly Detection Systems, Integrating Human Feedback, Mitigating Adversarial Effects.

UNIT 3 **Malware Analysis-** Understanding Malware: Defining Malware Classification, Feature generation: Data Collection, Generating Features, Feature Selection, From Features to Classification: How to Get Malware Samples and Labels

UNIT 4 **Network Traffic Analysis:** Access Control and Authentication, Intrusion Detection, Detecting In-Network Attackers, Data-Centric Security, Honeypots, Building a Predictive Model to Classify Network Attacks: Exploring the Data, Data Preparation, Classification,Supervised Learning, Semi-Supervised , Learning, Unsupervised Learning, Advanced Ensembling.

UNIT 5 **Production Systems**- Machine Learning System Maturity and Scalability, Data Quality: Bias in Datasets, Missing Data, Data quality; Model Quality: Hyperparameter Optimization, Feature: Feedback Loops, A/B Testing of Models, Repeatable and Explainable Results; Performance: Goal: Low Latency, High Scalability, Performance Optimization; Maintainability: Problem: Check pointing, Versioning, and Deploying Models, Graceful Degradation, Tuning and Configurable, Monitoring and Alerting; Security and Reliability: Robustness in Adversarial Contexts, Data Privacy Safeguards and Guarantees, Feedback and Usability

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Clarence Chio and David Freeman, Machine Learning & Security: Protecting Systems With Data And Algorithms Protecting Systems, Latest Edition, O'Reilly.
2. Mark Stamp, Introduction To Machine Learning With Applications In Information Security, CRC Press, Taylor & Francis Group.

REFERENCE BOOKS:

1. Marcus A. Maloof, Machine Learning and Data Mining for Computer Security Methods and Applications (Advanced Information and Knowledge Processing).
2. Tony Thomas, Athira P. Vijayaraghavan, Sabu Emmanuel, Machine Learning Approaches in Cyber Security Analytics, Springer.
3. Gupta, Brij Sheng, Quan Z, Machine learning for computer and cyber security principles, algorithms, and practices-CRC Press (2019).

RESEARCH JOURNALS:

1. IEEE Transactions on Information Forensics and Security
2. ACM Transactions on Cyber-Physical Systems

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-III)
CO 71722: SOFTWARE ARCHITECTURE

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Software Engineering.

COURSE OBJECTIVES: To provide a comprehensive skill for designing and implementation of enterprise application development using the software architectures.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Explain the quality requirements of a software system and fundamentals terminologies of software architecture.
2. Use fundamental principles, guidelines, architectural styles, patterns, techniques, and tools to design and analyze software architecture of the real word problems.
3. Implement software architecture to solve specified problems.
4. Appraise the quality of software architectures.

COURSE CONTENTS:
THEORY:

UNIT 1 Overview of Software development methodology and software quality model, different models of software development and their issues. Introduction to software architecture, software components and connectors, common software architecture frameworks, Architecture business cycle, architectural patterns, design patterns: Creational design patterns, Structural design patterns, Behavioral design patterns, Software quality attributes and models.

UNIT 2 Software Architectures styles: dataflow architecture, pipes and filters architecture, call- and return architecture, data-centered architecture, layered pattern architecture, agent based architecture, services and micro-services oriented architecture etc.

UNIT 3 Software architecture implementation technologies: Software Architecture Description Languages (ADLs), JSP, Servlets, struts, hibernate, EJBs; middleware: JDBC, JNDI, JMS and CORBA. Angular JS, Role of UML in software architecture.

UNIT 4 Software Architecture analysis and design: Architectural Tactics, requirements for architecture and the life-cycle view of architecture design and analysis methods, Cost Benefit Analysis Method (CBAM), Architecture Tradeoff Analysis Method (ATAM). Active Reviews for Intermediate Design (ARID), Attribute Driven Design method (ADD), architecture reuse, Domain –specific Software architecture.

UNIT 5 Software Architecture documentation: principles of sound documentation, refinement, context diagrams, variability, software interfaces. Documenting the behavior of software elements and software systems, documentation package using a seven-part template.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Bass, L., P. Clements, and R. Kazman, "Software Architecture in Practice", Second Edition, Prentice-Hall 2015.
2. Jim Keogh, "J2EE – Complete Reference", Tata McGraw Hill, 2008.
3. Dikel, David, D. Kane, and J. Wilson, "Software Architecture: Organizational Principles and Practices", Prentice-Hall, 2001.

REFERENCE BOOKS:

1. Bennett, Douglas, "Designing Hard Software: The Essential Tasks", Prentice-Hall, 1997.
2. Clements, Paul, R. Kazman, M. Klein, "Evaluating Software Architectures: Methods and Case Studies", Addison Wesley, 2001.
3. Albin, S. "The Art of Software Architecture", Indiana: Wiley, 2003.
4. Robert Mee, and Randy Stafford, "Patterns of Enterprise Application Architecture", Addison-Wesley, 2002.
5. Witt, B., T. Baker and E. Meritt, "Software Architecture and Design: Principles, Models and Methods", Nostrand Reinhold, 1994.

WEB SITES:

1. www.sei.cmu.edu/architecture/
2. www.softwarearchitectureportal.org/

RESEARCH JOURNALS:

1. IEEE Transactions on software engineering
2. IEEE Transactions on software

IEEE/ACM CONFERENCES:

1. Automated Software Engineering (ASE) Conference

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-III)
CO 71723: EMBEDDED SYSTEMS AND INTERNET OF THINGS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Computer Network.

COURSE OBJECTIVES:
To provide a foundational concept on embedded systems to the students to design and develop the basic IoT applications.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define various components and working of embedded systems.
2. Describe the functions of embedded operating systems like scheduling, memory management etc.
3. Explain the working of various types of sensors and actuators in different domains.
4. Design and develop basic applications in IoT.

COURSE CONTENTS:
THEORY:

UNIT 1 **Embedded Systems Overview:** Introduction, Definition, Characteristics & Salient Features, Classification, Overview of Embedded System Architecture & Recent Trends, Application Areas. Hardware Architecture: Basic building blocks, Devices and Communication Interfaces, Software Architecture: Embedded Operating System, Applications Software & Communication Software (Firmware).

UNIT 2 **ARMCortex-M3/M4 processor:** Introduction ARM architecture and Cortex: M series, Introduction to the Tiva family viz. TM4C123x and its targeted applications, Tiva block diagram, address space, on-chip peripherals, Simplified view: block diagram, programming model: Registers, Operation modes, Exceptions and Interrupts, Reset Sequence. ARM Cortex M4 Peripherals: I/O pin multiplexing, GPIO control, Watchdog Timer, System Clocks and control, Hibernation Module on TM4C, Introduction to Interrupts, Interrupt vector table, interrupt programming, Timers, ADCs, UART and other serial interfaces: USB, PWM, RTC, DMA, QEI on TM4C microcontroller.

UNIT 3 **Embedded Operating System:**General Purpose Operating Systems vs Embedded Operating Systems, Overview of Embedded Operating Systems, detailed study of Real Time Operating System(RTOS) such as TI RTOS, Task Scheduling, Task Synchronization, Inter- task communication concepts on Embedded TI RTOS Platform, Case Studies of Typical Embedded Systems: Digital Camera and Washing Machine.

UNIT 4 **Introduction to IoT:** Definition of IoT; IoT Layered Architecture; Introduction and working of different types of Sensors and actuators; Interfacing of sensors and actuators with ESP32/Arduino/Raspberry Pi etc.

UNIT 5 **IoT Protocols:** IoT communication protocols e.g. Bluetooth, BLE, Zigbee, Wi-fi etc. IoT Messaging protocols e.g. MQTT etc.; IoT Routing protocols e.g. RPL; CoAP; IoT and Cloud; Case studies Smart Home, Smart Farming etc.

DIRECT ASSESMENT:
ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Embedded Systems Architecture Programming and Design by Raj Kamal, II edition, Tata MC Graw-Hill.
2. Introduction to Embedded Systems, by Shibu K V, Tata MC Graw-Hill.
3. Introduction to Embedded Systems- A Cyber-Physical Systems Approach, by Lee &Seshia, MIT Press.
4. RajkumarBuyya, “Internet of Things – Principles and Paradigms", Elsevier Publication.

REFERENCE BOOKS:

1. Embedded Systems Design by Steve Heath, II edition, Newnes publications.
2. Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers by Tammy Noergaard, Elsevier.
3. Timothy Chou, “Precision: Principles, Practices and Solution for the Internet of Things.

RESEARCH JOURNALS:

1. ACM Transactions on Internet of Things
2. IEEE Internet of Things Journal

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-III)
CO 71724: HUMAN COMPUTER INTERACTION

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Programming skill**s.**

COURSE OBJECTIVES:
The main objective of this course is to Design, Implement and Evaluate effective, friendly and usable graphical computer Interfaces and to identify various techniques for interface analysis and design.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define the basics behind HCI design approaches and evaluate effective and usable graphical computer interfaces.
2. Apply an interactive design process and universal design principles for modeling HCI systems.
3. Design and Implement HCI approaches for solving real-life problems.
4. Analyze and discuss HCI issues in groupware, ubiquitous computing, virtual reality, multimedia, and Word Wide Web-related environments.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to Human Computer Interaction, HCI History, HCI Frameworks, HCI paradigms, Aspects of Cognition, Heuristic Evaluation, Usability Guidelines, Choosing Among Usability Methods.

UNIT 2 Introduction to Evaluation, Predictive Evaluation, User modeling, UCD Process, Usability Principles, User-centered Design, Interaction Styles, Higher Cognition: Metaphor, Widget Survey, Other Interaction Styles, Choosing among Interaction Styles

UNIT 3 Human Abilities, Predictive Models and Cognitive Models, Descriptive Cognitive models, Ubiquitous Computing, , Specifying and Prototyping: Low-Fidelity Prototyping, Transition Diagrams, Visual Basic Prototyping.

UNIT 4 Interface Implementation: Events and Handlers, The Model-View-Controller Design Pattern, Responsiveness Issues, Time-scales and the Illusion of Multi-Tasking, Natural language & Speech, Information visualization, tangible user Interfaces.

UNIT 5 Web, Mobile, Speech and Multimodal, Groupware, Games, etc, UI Software, UI Agents, Case studies: Windows Swings.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Designing the User Interface, 6 th Edition. Ben Shneiderman, Catherine Plaisant, et al., Addison Wesley, 2017.
2. Interaction Design: Beyond Human-Computer Interaction, Helen Sharp, Jennifer Preece, Yvonne Rogers, Wiley; 5 editions, 2019.
3. Doing Better Statistics in Human-Computer Interaction, Paul Cairns, Cambridge University Press, 2019.

REFERENCE BOOKS:

1. Human Computer Interaction, 3rd Edition, Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale Prentice Hall, 2004.
2. Designing Web Interfaces, M. Rees, A. White, B. White, Pearson, 2001.

RESEARCH JOURNALS:

1. ACM Transactions on Computer-Human Interaction.
2. AIS Transactions on Human-Computer Interaction.
3. Journal of Organizational and End User Computing.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-IV)
CO 71764: DATA SCIENCE & ANALYTICS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basics of statistics.

COURSE OBJECTIVES:
To enable students to understand data and analyze data for various learning algorithms.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Identify the characteristics of datasets and visualize the data.
2. Select and implement data preprocessing techniques that are suitable for the applications under consideration.
3. Solve problems associated with data characteristics such as high dimensionality, dynamically growing data
4. Apply and implement text mining algorithms and comprehend big data analytics.

COURSE CONTENTS:
THEORY:

UNIT 1 Data Wrangling and Exploratory Analysis, Data Transformation & Cleaning, Feature Extraction, Data Visualization. Introduction to contemporary tools and programming languages like R, Python etc. for data analysis.

UNIT 2 Statistical & Probabilistic analysis of Data: Multiple hypothesis testing, Parameter Estimation methods, Confidence intervals, Bayesian statistics and Data Distributions.

UNIT 3 Dimensionality reduction: PCA & SVD, Correlation & Regression analysis, Training & Testing data: Overfitting & Under fitting.

UNIT 4 Introduction to Information Retrieval: Boolean Model, Vector model, Probabilistic Model, Text based search: Tokenization , Tf-IDF, stop words and n-grams , synonyms and parts of speech tagging

UNIT 5 Introduction to Web Search & Big Data : Crawling and Indexes, Search Engine architectures, Link Analysis and ranking algorithms such as HITS and PageRank Hadoop File system & MapReduce Paradigm

DIRECT ASSESMENT:
ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Field Cady, "The Data Science Handbook" , 1/e ,2018,Publisher: Wiley.
2. Sinan Ozdemir, "Principles of Data Science ", 1/e, 2016Packt Publishing Limited.

REFERENCE BOOKS:

1. Peter Bruce, "Practical Statistics for Data Scientists: 50 Essential Concepts",Shroff/O'Reilly; First edition (2017).
2. Pang-Ning Tan, "Introduction to Data Mining", Pearson Edu., 2007.
3. Ricardo Baeza-Yates andBerthier Ribeiro-Neto, "Modern Information Retrieval", Pearson Education, 2004.

RESEARCH JOURNALS:

1. International Journal of Data Science and Analytics.
2. ACM Transactions on Knowledge Discovery in Data (TKDD).
3. Data Mining and Knowledge Discovery journal.
4. The international journal Advances in Data Analysis and Classification.

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-IV)
CO 71765: CYBER SECURITY AND FORENSICS

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Knowledge of cryptographic algorithms, Network Protocols.

COURSE OBJECTIVES:
To make students understand computer crimes and methods to investigate them.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define and explain computer crimes.
2. Assess threats and vulnerabilities in operating systems and computer networks and their security requirements
3. Plan and implement preliminary cyber forensic investigations.
4. Design a secure cyber system and comprehensive cyber security policy.

COURSE CONTENTS:
THEORY:

UNIT 1 Review of cryptographic security mechanisms, Definition and types of computer crimes, Distinction between computer crimes and conventional crimes, Types of computer crimes, Need for a Comprehensive Cyber Security Policy, Man-in-the-Middle Attacks

UNIT 2 Review of Network Security: Threats on World Wide Web: phishing, Format string Attack, URL manipulation attack. Cross-site Scripting Attack, SQL Injection Attack etc.; Scrutinizing E-mail, Validating E-mail header information, Tracing Internet access, Browser Helper Objects

UNIT 3 Security in Operating Systems, Memory and Address protection, File system protection, Cases- Windows, Linux O/s. DLL Injection, Virtual Machine Detection

UNIT 4 Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation; Introduction to Database security methods, Attacks against Privileged User Accounts and Escalation of Privileges

UNIT 5 Memory Forensics, Conducting disk-based analysis, Investigating Information-hiding, Tracing memory in real-time, Memory Analysis Frameworks, Dumping Physical Memory, Installing and Using Volatility, Finding Hidden Processes, Volatility Analysis

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. SunitBelapure, Nina Godbole, “Cyber Security”, Latest edition, Wiley.
2. Bill Nelson, Amelia Phillips, Christopher Steuart, “Guide to Computer Forensics and Investigations”, Latest edition, Cengage Learning.
3. James Graham, Richard Howard, Ryan Olson, “Cyber Security Essentials”, CRC Press, Tayor And Francis Group.

REFERENCE BOOKS:

1. Michael Hale Ligh, Steven Adair Blake Hartstein, Matthew Richard, Malware Analyst’s Cookbook and DVD, Tools and Techniques for Fighting Malicious Code, Wiley Publishing Inc.

MOOCS COURSES ON SWAYAM:

1. Digital Forensic, By Dr. Navjot Kaur Kanwal,
2. Cyber Security By Prof. Padmavathi G
3. Ethical Hacking By Prof. Indranil Sen Gupta

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-IV)
CO 71766: BIG DATA

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: BasicMathematics and Statistics, Data Science and Engineering.

COURSE OBJECTIVES: Students will be able to understand the challenges of Big data and gain knowledge about Hadoop Eco system and develop applications using data processing and analytics to solve real world problems.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Explain the concept and challenges of Big Data and Big Data Analytics.
2. Examine Hadoop Eco System and develop Big Data Solutions using it.
3. Experiment and evaluate various large-scale analytics tools.
4. Analyze social networks graphs using mining techniques.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to Big data, Big data characteristics, Types of big data, Traditional versus Big data, Evolution of Big data, challenges with Big Data, Technologies available for Big Data, Infrastructure for Big data, Use of Data Analytics, Desired properties of Big Data system.

UNIT 2 Introduction to Hadoop, Core Hadoop components, Hadoop Eco system, Hive Physical Architecture, Hadoop limitations, RDBMS Versus Hadoop, Hadoop Distributed File system, Processing Data with Hadoop, Mapreduce Programming, Managing Resources and Application with Hadoop YARN, Apache Spark.

UNIT 3 Introduction to Hive Hive Architecture, Hive Data types, Hive Hive Query Language, Introduction to Pig, Anatomy of Pig, Pig on Hadoop, Use Case for Pig, ETL Processing, Data types in Pig running Pig, Execution model of Pig, Operators, Evalfunction,Data types of Pig.

UNIT 4 Introduction to NoSQL, NoSQL Business Drivers, NoSQL Data architectural patterns, Variations of NOSQL architectural patterns using NoSQL to Manage Big Data.

UNIT 5 Mining social Network Graphs: Introduction Applications of social Network mining, Social Networks as a Graph, Types of social Networks, Clustering of social Graphs Direct Discovery of communities in a social graph.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. RadhaShankarmani, M. Vijaylakshmi, " Big Data Analytics", Wiley, Second edition.
2. Seema Acharya, SubhashiniChellappan, " Big Data and Analytics", Wiley, First edition.

REFERENCE BOOKS:

1. Kai Hwang, Geoffrey C., Fox. Jack, J. Dongarra, “Distributed and Cloud Computing”, Elsevier, First edition.
2. Michael Minelli, Michele Chambers, AmbigaDhiraj, “Big Data Big Analytics”, Wiley.

RESEARCH JOURNALS:

1. IEEE Transactions on BIG DATA
2. Springer Journal Big Data
3. ACM transaction on Data science

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-IV)
CO 71767: BLOCK CHAIN

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Basic knowledge of computer security and networking.

COURSE OBJECTIVES: Students will be able to understand the features, challenges and solutions of cryptocurrency, blockchains, and cryptography.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Define the basics of cryptocurrency, blockchains, and cryptography.
2. Compare and contrast different blockchain technologies.
3. Apply the concepts and techniques of blockchain in computer security.
4. Identify the shortcomings of blockchain technology and their corresponding solutions.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction: Need for Distributed Record Keeping, Modeling faults and adversaries Byzantine Generals problem, Consensus algorithms and their scalability problems, Why Nakamoto Came up with Blockchain based cryptocurrency? Technologies Borrowed in Blockchain – hash pointers, consensus, byzantine fault-tolerant distributed computing, digital cash etc.
Basic Distributed Computing: Atomic Broadcast, Consensus, Byzantine Models of fault tolerance
Basic Crypto primitives: Hash functions, Puzzle friendly Hash, Collison resistant hash, digital signatures, public key crypto, verifiable random functions, Zero-knowledge systems.

UNIT 2 Blockchain 1.0: Bitcoinblockchain, the challenges, and solutions, proof of work, Proof of stake, alternatives to Bitcoin consensus, Bitcoin scripting language and their use.

UNIT 3 Blockchain 2.0: Ethereum and Smart Contracts, The Turing Completeness of Smart Contract Languages and verification challenges, Using smart contracts to enforce legal contracts, comparing Bitcoin scripting vs. Ethereum Smart Contracts.

UNIT 4 Blockchain 3.0: Hyperledger fabric, the plug and play platform and mechanisms in permissioned blockchain.

UNIT 5 Privacy, Security issues in Blockchain: Pseudo-anonymity vs. anonymity, Zcash and Zk-SNARKS for anonymity preservation, attacks on Blockchains – such as Sybil attacks, selfish mining, 51% attacks - -advent of algorand, and Sharding based consensus

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016. (Free download available).
2. Blockchain Science: Distributed Ledger Technologies, Rogen Wattenhofer, Inverted Forest Publishing, 2019.

REFERENCE BOOKS:

1. Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015.
2. J.A.Garay et al, The bitcoin backbone protocol - analysis and applications EUROCRYPT 2015 LNCS VOl 9057, (VOLII), pp 281-310. (Also available at eprint.iacr.org/2016/1048).
3. R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks, EUROCRYPT 2017, (eprint.iacr.org/2016/454).
4. R.Pass et al, Fruitchain, a fair blockchain, PODC 2017 (eprint.iacr.org/2016/916).

DEPARTMENT OF COMPUTER ENGINEERING
M. TECH. I YEAR (2YDC)
SEMESTER-B (ELECTIVE-IV)
CO 71768: HIGH PERFORMANCE COMPUTING

*Hours per Week			Th. Credit	Pr. Credit	MAXIMUM MARKS				
L	T	P			TH	CW	SW	Pr.	Total
3	-	-	3	-	70	30	-	-	100

PRE-REQUISITES: Computer Architecture, Operating Systems.

COURSE OBJECTIVES: Students will understand the architectural fundamentals and design of shared memory multicore systems.

COURSE OUTCOMES:
After completing the course student should be able to:

1. Describe different high performance computing architectures and evaluate their performance.
2. Compare and contrast memory/processor performance by applying various optimization techniques.
3. Evaluate various proposed changes in different types of architecture, instructions set and memory using gem5 simulator.
4. Define pipelined processors, identify various hazards and select solutions for them.

COURSE CONTENTS:
THEORY:

UNIT 1 Introduction to modern processors-: General Purpose cache based architecture-performance metric and benchmarks, Moore's Law, pipelining, super clarity, SIMD. Memory Hierarchies, Multicore processors, Multi-threaded processors,Basic Optimizations for serial codes: - Scalar profiling, common sense optimizations, Simple measures and their impacts, role of compilers, C++ optimizations.MIPS Assembly Language Program, Memory layout, SPIM Simulator.

UNIT 2 Superscalar architecture: An overview, Instruction flow optimization: Handling branches, Branch predictors, Advanced optimization in instruction flow, register flow techniques: Register renaming and out of order execution, Out of order execution, Advanced data flow techniques: Instruction reuse and value prediction, Memory data flow, Advanced memory data flow architectures, Limits of superscalar architectures

UNIT 3 Beyond ILP, Multi-threading, Simultaneous multithreaded (SMT) architectures, SMT architecture: Choices, SMT performance on various designs, SMT architecture: OS impact and adaptive architectures.

UNIT 4 VLIW architectures, Multiscalar architecture, Multi-core Architectures, Multicore Interconnect – NOC Network-on-Chip,Tiled Chip Multicore Processors(TCMP), Routing Techniques in Network on Chip(NoC), NoC Router Microarchitecture, TCMP and NoC: Design and Analysis.

UNIT 5 Super-scalar and Memory Hierarchy Introduction Basic and Advanced Optimization Techniques in Cache Memory, Cache Optimization using gem5, Cache Coherence, Cache Consistency model**.**Case Study: Locking hardware and algorithm, Task Scheduling on Multicore System GP-GPU Architecture, CPU-GPU Integration.

DIRECT ASSESMENT:

ASSESMENT OF THEORY-

1. Internal Assessment for continuous evaluation, mid-term tests, Tutorials, Quizzes, Class Performance, etc. (30%).
2. End semester Theory Exam (70%).

INDIRECT ASSESMENT:

1. Feedback of students on attainment of cos.
2. Feedback of students on classroom learning.
3. External examiners feedback on Cos.

TEXT BOOKS RECOMMENDED:

1. George Hager and Gerhard Wellein, “Introduction to high performance Computing for scientists and engineers”, CRC Press.
2. Charles Severance, Kevin Dowd, “High Performance Computing”, 2nd Edition, O'Reilly.
3. Patterson, D.A., and Hennessy, J.L., “Computer Organization and Design: The Hardware/Software Interface”, Morgan Kaufmann Publishers, 4th Edition, Inc.2005.
4. Patterson, D.A., and Hennessy, J.L., “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann Publishers, 4th Edition, Inc.2005.
5. Hamacher, V.C., Vranesic, Z.G., and Zaky, S.G., “Computer Organization”, 5/e. McGraw-Hill. 2008.
6. Tanenbaum, A.S, “Structured Computer Organization”, Prentice-Hall of India. 1994.

REFERENCE BOOKS:

1. Stalling W, “Computer Organization and Architecture”, Pearson Education India, 2008.
2. D V Hall, Microprocessors and Interfacing, TMH, 1995.
3. Brey. Barry B, The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing, Prentice Hall India, 2005.

REFERENCE JOURNALS:

1. Inderscience International Journal of Electronic Government
2. Springer CCF Transactions on High Performance Computing
3. ACM SIGHPC - Special Interest Group on High Performance Computing
4. International Journal of Computer Information Systems and Industrial Management Applications, ISSN 2150-7988, [IJCISIM]