VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH (COMPUTER ENGINEERING)

I Semester **Total Credits: 23**

Subject Code	Name of the Subject	Teaching hours/week	Practical / Field Work / Assignment/ Tutorials	Duration of Exam in Hours	Marks for		Total Marks	Credits
					I.A.	Exam		
14SCE11	Advanced Digital Design	4	--	03	50	100	150	4
14SCE12	Cloud Computing	4	2 *	03	50	100	150	4
14SCE13	Embedded Computing Systems	4	2 *	03	50	100	150	4
14SCE14	Advances in Computer Architecture	4	2	03	50	100	150	4
14SCE15x	Elective – I	4	2	03	50	100	150	4
14SCE16	Advanced Digital Design Laboratory	--	3	03	25	50	75	2
14SCE17	Seminar #	--	3	--	25	--	25	1
Total		**20**	**13**	**18**	**300**	**550**	**850**	**23**

Elective I

14SCE151 Computer Systems Performance Analysis
14SCE152 Distributed Operating System
14SCE153 Software Agents
14SCE154 Bio-Informatics

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH (COMPUTER ENGINEERING)

II Semester **Total Credits: 23**

Subject Code	Name of the Subject	Teaching hours/week	Practical / Fieldwork / Assignment / Tutorials	Duration of Exam in Hours	Marks for		Total Marks	CREDITS
					I.A.	Exam		
14SCE21	Managing Big Data	4	2 *	03	50	100	150	4
14SCE22	Mobile Application Development	4	2 *	03	50	100	150	4
14SCE23	Wireless Networks & Mobile Computing	4	--	03	50	100	150	4
14SCE24	Multi Core Architecture and Programming	4	2	03	50	100	150	4
14SCE25x	Elective – II	4	2	03	50	100	150	4
14SCE26	Wireless Networks & Mobile Computing Laboratory	--	3	03	25	50	75	2
14SCE27	Seminar #	--	3	--	25	--	25	1
	** Project Phase I (6 Week Duration)	-- -- --	--	--	--	--	--	--
Total		**20**	**13**	**18**	**300**	**550**	**850**	**23**

ELECTIVE- II

14SCE251 Data Mining & Data Warehousing
14SCE252 Pattern Recognition
14SCE253 Advances in Storage Area Network
14SCE254 Decision Support System

**** Between the II Semester and III Semester after availing a vacation of 2 weeks.**

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH. COMPUTER ENGINEERING

III Semester: INTERNSHIP **Total Credits: 20**

Subject Code	Name of the Subject	No. of Hrs./Week		Duration of the Exam in Hours	Marks for		Total Marks	CREDITS
		Lecture	Practical/Field Work		I.A.	Exam		
14SCE31	Seminar / Presentation on Internship (After 8 weeks from the date of commencement)	--	--	-	25	--	25	1
14SCE32	Report on Internship	--	--	-	--	75	75	15
14SCE33	Evaluation and Viva-voce	--	--	3	--	50	50	4
	Total	**-**	**--**	**3**	**25**	**125**	**150**	**20**

***The student shall make a midterm presentation of the activities undertaken during the first 8 weeks of internship to a panel comprising Internship Guide, a senior faculty from the department and Head of the Department.**

 # The College shall facilitate and monitor the student internship program.

The internship report of each student shall be submitted to the University.

****Between the III Semester and IV Semester after availing a vacation of 2 weeks.**

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SCHEME OF TEACHING AND EXAMINATION FOR
M.TECH(COMPUTER ENGINEERING)

IV Semester **Total Credits: 28**

Subject Code	Name of the Subject	Teaching hours/week		Duration of Exam in Hours	Marks for		Total Marks	CREDITS
		Lecture	Field work/ Assignment/ Tutorials		I.A.	Exam		
14SCE41	ARM Processors *	4	2 *	03	50	100	150	4
14SCE42x	Elective-III	4	--	03	50	100	150	4
14SCE43	Evaluation of Project Phase-II	0	--	0	25	--	25	1
14SCE44	Evaluation of Project Phase-III	0	--	0	25	--	25	1
14SCE45	Evaluation of Project Work and Viva-voce	--	--	03	--	100 + 100	200	18
Total		**8**	**--**	**09**	**150**	**400**	**550**	**28**
Grand Total (I to IV Sem.)		**Marks 2400 ; Total Credits: 94**						

Elective – III

14SCE421 Wireless Adhoc Networks
14SCE422 Wireless Sensor Network
14SCE423 Optical Networks
14SCE424 Enterprise Application Programming

L- Lecture, T- Tutorial, P- Practical

Note:

***Lab Classes for these Core Subjects are Compulsory (Practical will be Evaluated for 20 marks and Internal assessment for 30 marks). Lab journals Should be Maintained.**

 # Seminar: Topics should be chosen from IEEE/ACM/Elsevier/Springer/any Refereed - Journals /Transactions. Encourage students to convert these seminar topics into a good survey paper or Technical paper.

1).Project Phase – I: 6 weeks duration shall be carried out between II and III Semester. Candidates in consultation with guide shall carryout literature survey / visit to Industries to finalize the topic of dissertation.

2) Internship:- 24 weeks Duration in 3rd Semester, Evaluation of Marks - Presentation : 25 marks, Report writing and Submission :75 marks and At the end of Internship Viva-Voce Exams shall be conducted for 50 marks.

3).Project Work: 20 weeks duration in IV Semester carries total marks of 250.

4)Project Phase II: 4 days for project work in a week during IV Semester. Evaluation shall be taken during the 8th week of the IV Semester. Total Marks shall be 25.

5).Project Phase – III : Evaluation shall be taken up at the end of the IV Semester for 25 marks. After the Project report is submitted, Project Work Evaluation and Viva-Voce Examination shall be conducted. Total Marks Shall be 50+50+100=200 (50 Marks for Internal Guide, 50 Marks for External and 100 for Viva-Voce).

Marks of Evaluation of Project:

I) The I.A. Marks of Project Phase – II & III shall be sent to the University along with Project Work report at the end of the Semester.

II) The Project Valuation and Viva-Voce will be conducted by a committee consisting of the following:

a) Head of the Department (Chairman)
b) Guide
c) Two Examiners appointed by the university.(out of two external examiners at least one should be present).

SEM I **Year 2014-15**

Course Title: : Advanced Digital Design	**Course Code: 14SCE11**
Credits(L:T:P): 4:0:0	**Core/Elective: Core**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To learn about various IC technology options
- To learn Logic simulation, Design verification, Verilog.
- To understand Behavioral modeling, Boolean-Equation, Flip-Flops and Latches; multiplexers, encoders, and decoders, synchronizers for asynchronous signals.
- To understand combinational logic; three-state devices and bus interfaces; Registered logic; registers and counters; Resets; Divide and conquer: Partitioning a design.
- Basics of PLA; PAL; Programmability of PLDs; CPLDs; FPGAs;

TOPICS:

MODULE I

Introduction: Design methodology – An introduction; IC technology options. **10 Hours**

MODULE II

Logic Design with Verilog: Structural models of combinational logic; Logic simulation, Design verification, and Test methodology; Propagation delay; Truth-Table models of Combinational and sequential logic with Verilog. **10 Hours**

MODULE III

Logic Design with Behavioral Models: Behavioral modeling; A brief look at data types for behavioral modeling; Boolean-Equation – Based behavioral models of combinational logic; Propagation delay and continuous assignments; Latches and Level – Sensitive circuits in Verilog; Cyclic behavioral models of Flip-Flops and Latches; Cyclic behavior and edge detection; A comparison of styles for behavioral modeling; Behavioral models of multiplexers, encoders, and decoders; Dataflow models of a Linear-Feedback Shift Register; Modeling digital machines with repetitive algorithms; Machines with multi-cycle operations; Design documentation with functions and tasks; Algorithmic state machine charts for behavioral modeling; ASMD charts; Behavioral models of counters, shift registers and register files; Switch debounce, meta-stability and synchronizers for asynchronous signals; Design example.

10 Hours

MODULE IV

Synthesis of Combinational and Sequential Logic: Introduction to synthesis; Synthesis of combinational logic; Synthesis of sequential logic with latches; Synthesis of three-state devices and bus interfaces; Synthesis of sequential logic with flip-flops; Synthesis of explicit state machines; Registered logic; State encoding; Synthesis of implicit state machines, registers and counters; Resets; Synthesis of gated clocks and clock enables; Anticipating the results of synthesis; Synthesis of loops; Design traps to avoid; Divide and conquer: Partitioning a design.

10 Hours

MODULE V

Programmable Logic and Storage Devices: Programmable logic devices; Storage devices; PLA; PAL; Programmability of PLDs; CPLDs; FPGAs; Verlog-Based design flows for FPGAs; Synthesis with FPGAs. **10 Hours**

Course Outcomes:

The students shall be able to:

- Work on various IC technology options.
- Implement Logic simulation, Design verification, Verilog.
- Work on Flip-Flops and Latches; multiplexers, encoders, and decoders, synchronizers for asynchronous signals.
- Design and implement circuits on combinational logic; Registered logic; registers and counters; Resets; Divide and conquer: Partitioning a design.

TEXT BOOKS:

1. Michael D. Celetti: Advanced Digital Design with the Verilog HDL, PHI, 2013.

REFERENCE BOOKS:

1. PeterJ. Asheden: Degital Design –An Embedded Systems Approach Using VERILOG, ELSEVIER 2013.
2. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic with Verilog Design, Tata Mc-Graw Hill 2009.

SEM I **Year 2014-15**

Course Title: : Cloud Computing	**Course Code: 14SCE12**
Credits(L:T:P): 3:0:1	**Core/Elective: Core**
Type of Course: Lecture & Practical	**Total Contact Hours:50**

COURSE OBJECTIVES

- To learn how to use Cloud Services.
- To implement Virtualization
- To implement Task Scheduling algorithms.
- Apply Map-Reduce concept to applications.
- To build Private Cloud.

TOPICS:

MODULE I

Introduction, Cloud Infrastructure: Cloud computing, Cloud computing delivery models and services, Ethical issues, Cloud vulnerabilities, Cloud computing at Amazon, Cloud computing the Google perspective, Microsoft Windows Azure and online services, Open-source software platforms for private clouds, Cloud storage diversity and vendor lock-in, Energy use and ecological impact, Service level agreements, User experience and software

licensing. Exercises and problems.

10 Hours

MODULE II

Cloud Computing: Application Paradigms.: Challenges of cloud computing, Architectural styles of cloud computing, Workflows: Coordination of multiple activities, Coordination based on a state machine model: The Zookeeper, The Map Reduce programming model, A case study: The Gre The Web application, Cloud for science and engineering, High-performance computing on a cloud, Cloud computing for Biology research, Social computing, digital content and cloud computing.

10 Hours

MODULE III

Cloud Resource Virtualization: Virtualization, Layering and virtualization, Virtual machine monitors, Virtual Machines, Performance and Security Isolation, Full virtualization and paravirtualization, Hardware support for virtualization, Case Study: Xen a VMM based paravirtualization, Optimization of network virtualization, vBlades, Performance comparison of virtual machines, The dark side of virtualization, Exercises and problems.

10 Hours

MODULE IV

Cloud Resource Management and Scheduling: Policies and mechanisms for resource management, Application of control theory to task scheduling on a cloud, Stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds, Coordination of specialized autonomic performance managers, A utility-based model for cloud-based Web services, Resourcing bundling: Combinatorial auctions for cloud resources, Scheduling algorithms for computing clouds, Fair queuing, Start-time fair queuing, Borrowed virtual time, Cloud scheduling subject to deadlines, Scheduling MapReduce applications subject to deadlines, Resource management and dynamic scaling, Exercises and problems.

10 Hours

MODULE V

Cloud Security, Cloud Application Development: Cloud security risks, Security: The top concern for cloud users, Privacy and privacy impact assessment, Trust, Operating system security, Virtual machine Security, Security of virtualization, Security risks posed by shared images, Security risks posed by a management OS, A trusted virtual machine monitor, Amazon web services: EC2 instances, Connecting clients to cloud instances through firewalls, Security rules for application and transport layer protocols in EC2, How to launch an EC2 Linux instance and connect to it, How to use S3 in java, Cloud-based simulation of a distributed trust algorithm, A trust management service, A cloud service for adaptive data streaming, Cloud based optimal FPGA synthesis .Exercises and problems. **10 Hours**

LAB EXPERIMENTS

NOTE: Simulate using object oriented programming, any available cloud environment **(Eg; Amazon cloud) and VM ware for resource virtualization.**

1. Create a Collaborative learning environment for a particular learning topic using Google Apps. Google Drive, Google Docs and Google Slides must be used for hosting e-books, important articles and presentations respectively. The instructor must use the Google Sheets to convey the timetable for different events and for analyzing the scores for individual assignment submission.

2. Modeling and simulation Cloud computing environments, including Data Centers, Hosts and Cloudlets and perform VM provisioning using CloudSim: Design a host with two CPU cores, which receives request for hosting two VMs, such that each one requires two cores and plans to host four tasks units. More specifically, tasks t1, t2,

t3 and t4 to be hosted in VM1, while t5, t6, t7, and t8 to be hosted in VM2. Implement space-shared allocation policy and time-shared allocation policy. Compare the results.

3. Model a Cloud computing environment having Data center that had 100 hosts. The hosts are to be modeled to have a CPU core (1000 MIPS), 2 GB of RAM and 1 TB of storage. Consider the workload model for this evaluation included provisioning requests for 400 VMs, with each request demanding 1 CPU core (250 MIPS), 256 MB of RAM and 1 GB of storage. Each VM hosts a *web-hosting application service*, whose CPU utilization distribution was generated according to the uniform distribution. Each instance of a webhosting service required 150,000 MIPS or about 10 minutes to complete execution assuming 100% utilization. Simulate Energy-conscious model for power consumption and power management techniques such as Dynamic Voltage and Frequency Scaling (DVFS). Initially, VMs are to be allocated according to requested parameters (4 VMs on each host). The Cloud computing architecture that is to be considered for studying energy conscious resource management techniques/policies included a data center, CloudCoordinator, and Sensor component. The CloudCoordinator and Sensor perform their usual roles. Via the attached Sensors (which are connected with every host), CloudCoordinator must periodically monitor the performance status of active VMs such as load conditions, and processing share. This real time information is to be passed to VMM, which can use it for performing appropriate resizing of VMs and application of DVFS and soft scaling. CloudCoordinator continuously1 has to adapt allocation of VMs by issuing VM migration commands and changing power states of nodes according to its policy and current utilization of resources.

4. Model and simulate the environment consisting of a data center with 10,000 hosts where each host was modeled to have a single CPU core (1200MIPS), 4GB of RAM memory and 2TB of storage. Consider the provisioning policy for VMs as space-shared, which allows one VM to be active in a host at a given instance of time. Make a request from the end-user (through the DatacenterBroker) for creation and instantiation of 50 VMs that had following constraints: 1024MB of physical memory, 1 CPU core and 1GB of storage. The application granularity was modeled to be composed of 300 task units, with each task unit requiring 1,440,000 million instructions (20 minutes in the simulated hosts) to be executed on a host. Minimal data transfer (300 KB) overhead can be considered for the task units (to and from the data center). After the creation of VMs, task units were submitted in small groups of 50 (one for each VM) at inter-arrival delay of 10 minutes.

5. Implement Map Reduce concept for
a. Strassen's Matrix Multiplication for a huge matrix.
b. Computing the average number of citation index a researcher has according to age among some 1 billion journal articles. 17. Consider a network of entities and relationships between them. It is required to calculate a state of each entity on the basis of properties of the other entities in its neighborhood. This state can represent a distance to other nodes, indication that there is a neighbor with the certain properties, characteristic of neighborhood density and so on. A network is stored as a set of nodes and each node contains a list of adjacent node IDs. Mapper emits messages for each node using ID of the adjacent node as a key. Reducer must recompute state and rewrite node with the new state. Implement this scenario.

COURSE OUTCOMES:

The student shall be able to:

- Compare the strengths and limitations of cloud computing
- Identify the architecture, infrastructure and delivery models of cloud computing
- Apply suitable virtualization concept.
- Choose the appropriate cloud player
- Address the core issues of cloud computing such as security, privacy and interoperability
- Design Cloud Services
- Set a private cloud

TEXT BOOK:

1. Dan C Marinescu: Cloud Computing Theory and Practice. Elsevier(MK) 2013.

REFERENCES:

1. Rajkumar Buyya , James Broberg, Andrzej Goscinski: Cloud Computing Principles and Paradigms, Willey 2014.
2. John W Rittinghouse, James F Ransome:Cloud Computing Implementation, Management and Security, CRC Press 2013.

SEM I **Year 2014-15**

Course Title: : Embedded Computing Systems	**Course Code: 14SCE13**
Credits(L:T:P): 3:0:1	**Core/Elective: Core**
Type of Course: Lecture & Practical	**Total Contact Hours:50**

COURSE OBJECTIVES

- To Provide a general overview of Embedded Systems
- To Show current statistics of Embedded Systems
- To Design a complete microprocessor-based hardware system
- To Design, code, compile, and test real-time software
- To Integrate a fully functional system including hardware and software
- To Gain the ability to make intelligent choices between hardware/software tradeoffs

TOPICS:

MODULE I

Introduction to embedded systems: Embedded systems, Processor embedded into a system, Embedded hardware units and device in a system, Embedded software in a system, Examples of embedded systems, Design process in embedded system, Formalization of system design, Design process and design examples, Classification of embedded systems, skills required for an embedded system designer.

07 Hours

MODULE II

Devices and communication buses for devices network: IO types and example, Serial communication devices, Parallel device ports, Sophisticated interfacing features in device ports, Wireless devices, Timer and counting devices, Watchdog timer, Real time clock, Networked embedded systems, Serial bus communication protocols, Parallel bus device protocols-parallel communication internet using ISA, PCI, PCI-X and advanced buses, Internet enabled systems-network protocols, Wireless and mobile system protocols. **13 Hours**

MODULE III

Device drivers and interrupts and service mechanism: Programming-I/O busy-wait approach without interrupt service mechanism, ISR concept, Interrupt sources, Interrupt servicing (Handling) Mechanism, Multiple interrupts, Context and the periods for context switching, interrupt latency and deadline, Classification of processors interrupt service mechanism from Context-saving angle, Direct memory access, Device driver programming. **10 Hours**

MODULE IV

Inter process communication and synchronization of processes, Threads and tasks: Multiple process in an application, Multiple threads in an application, Tasks, Task states, Task and Data, Clear-cut distinction between functions. ISRS and tasks by their characteristics, concept and semaphores, Shared data, Inter-process communication, Signal function, Semaphore functions, Message Queue functions, Mailbox functions, Pipe functions, Socket functions, RPC functions. **10 Hours**

MODULE V

Real-time operating systems: OS Services, Process management, Timer functions, Event functions, Memory management, Device, file and IO subsystems management, Interrupt routines in RTOS environment and handling of interrupt source calls, Real-time operating systems, Basic design using an RTOS, RTOS task scheduling models, interrupt latency and response of the tasks as performance metrics, OS security issues. Introduction to embedded software development process and tools, Host and target machines, Linking and location software.

10 Hours

LAB EXPERIMENTS:

MICROCONTROLLER AND EMBEDDED SYSTEM DESIGN

1. To get in touch with development tool/environment for ATMEL microcontroller program and architecture. To know the overview of Kiel software and an introduction to ATMEL 8051 architecture.
2. Write an embedded C program to add subtract multiply divide 16 bit data by ATMEL microcontroller. Write a separate module for each of the arithmetic module and bind it under a single module.
3. Write embedded c program to generate 10 KHz frequency using interrupts on P1.2 ant to view it on the CRO.
4. Write a program to interface 16X2 LCD to ATMEL microcontroller and use port P0 for interfacing it and use port P1 to interface key board.
5. Write a program to control DC motor using PWM method. To monitor the PWM status and control the speed of DC motor in 100% and 25% duty cycle pulse.
6. Write a program to control Position of servo motor. Using any of the ports to be input and output ports and provide an option for a switch to control the position of the motor.
7. Transmission and reception of data. The module has to be designed to have a clear understanding of how serial and parallel interface devices are controlled and interfaced with microcontroller.
8. To program and implement the temperature and pressure measurement units. Using appropriate sensor modules interfaced to the microcontroller indicate the changes in real world through the LEDs.

   NOTE; Use AT89C52 microcontroller as main kit with peripherals and Keil µVision 4/ Equivalent tool.

COURSE OUTCOMES:

The student shall be able to:

- Distinguish the characteristics of embedded computer systems.
- Examine the various vulnerabilities of embedded computer systems.
- Design an embedded system.
- Design and develop modules using RTOS.
- Implement RPC, threads and tasks

TEXT BOOKS:

1. **Raj Kamal**, "Embedded Systems: Architecture, Programming, and Design" 2nd edition , Tata McGraw hill-2013

Chapters: Chapter 1.1 to 1.5, 1.8 to 1.12, Chapter 3, 4, 7, 8 and 13.1 to 13.3.

References:

1. **Marilyn Wolf**, "Computer as Components, Principles of Embedded Computing System Design" 3rd edition, Elsevier-2014.

SEM I **Year 2014-15**

Course Title: : Advances in Computer Architecture	**Course Code: 14SCE14**
Credits(L:T:P): 4:0:0	**Core/Elective: Core**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To understand the recent trends in the field of Computer Architecture and identify performance related parameters
- To gain knowledge on pipelining.
- To gain knowledge on thread –level parallelism
- To gain insight on to the Memory hierarchy design

TOPICS:

MODULE I

Data-Level Parallelism in vector, SIMD, and GPU Architectures: Introduction, Vector Architecture, SIMD Instructions Set Extensions for Multimedia, Graphics Processing Units, Detecting and Enhancing Loop-level Parallelism, Crosscutting Issues, Putting it All Together: Mobile versus Server GPUs and Tesla versus Core i7, Fallacies and Pitfalls, Concluding Remarks, Historical Perspective and References Case Study and Exercises by Jason D. Bakos. **10 Hours**

MODULE II

Thread-Level Parallelism: Introduction, Centralized Shared-Memory Architectures, Performance of Symmetric Shared-Memory Multiprocessors, Distributed Shared-Memory and Directory-Based Coherence, Synchronization: The Basics, Models of Memory Consistency: An Introduction, Crosscutting Issues, Putting it All Together: Multicore Processors and Their Performance, Fallacies and Pitfalls, Concluding Remarks, Historical Perspective and References Case Studies and Exercises by Amr Zaky and David A. Wood.

10 Hours

MODULE III

Warehouse-Scale Computers to Exploit Request-Level and Data-Level Parallelism: Introduction, Programming Models and Workloads for Warehouse-Scale Computers, Computer Architecture of Warehouse-Scale Computers, Physical Infrastructure and Costs of Warehouse-Scale Computers, Cloud Computing: the Return of Utility Computing, Crosscutting Issues, Putting it All Together: A Google Warehouse-Scale Computer, Fallacies and Pitfalls, Concluding Remarks, Historical Perspective and References Case Studies and Exercises by Parthasarathy Ranganathan. **10 Hours**

MODULE IV

Vector Processors in More Depth : Why Vector Processors?, Basic Vector Architecture, Two Real-World Issues: Vector Length and Stride, Enhancing Vector Performance, Effectiveness of Compiler Vectorization, Putting it All Together: Performance of Vector Processors, a Modern Vector Supercomputer: The Cray X1 Fallacies and Pitfalls, Concluding Remarks, Historical Perspective and References Exercises.

10 Hours

MODULE V

Hardware and Software for VLIW and EPIC: Introduction: Exploiting Instruction-Level Parallelism Statically, Detecting and Enhancing Loop-Level Parallelism, Scheduling and Structuring Code for Parallelism, Hardware Support for Exposing Parallelism: Predicated Instructions, Hardware Support for Compiler Speculation, The Intel IA-64 Architecture and Itanium Processor, Concluding Remarks.

10 Hours

COURSE OUTCOMES:

The student shall be able to:

- Implement Pipelining concepts
- Identify the limitations of ILP
- Demonstrate an ability to apply theory and techniques to unseen problems.
- Understand the thread –level parallelism concepts.
- Explain concepts of vector process super computers and Cray X1.

TEXT BOOK:

1. Hennessey and Patterson: "Computer Architecture A Quantitative Approach", 5th Edition, Elsevier, 2013.

Reference Books:

1. Kai Hwang: Advanced Computer Architecture - Parallelism, Scalability, Programmability, 2nd Edition, Tata McGraw Hill, 2013.

SEM I **Year 2014-15**

Course Title: : Computer Systems Performance Analysis	**Course Code: 14SCE151**
Credits(L:T:P): 4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To understand the mathematical foundations needed for performance evaluation of computer systems
- To understand the metrics used for performance evaluation
- To understand the analytical modeling of computer systems
- To enable the students to develop new queuing analysis for both simple and complex systems
- To introduce the students to analytical techniques for evaluating scheduling policies

TOPICS:

MODULE I

Introduction: The art of Performance Evaluation; Common Mistakes in Performance Evaluation, A Systematic Approach to Performance Evaluation, Selecting an Evaluation Technique, Selecting Performance Metrics, Commonly used Performance Metrics, Utility Classification of Performance Metrics, Setting Performance Requirements. **10 Hours**

MODULE II

Workloads, Workload Selection and Characterization: Types of Workloads, addition instructions, Instruction mixes, Kernels; Synthetic programs, Application benchmarks, popular benchmarks. Work load Selection: Services exercised, level of detail; Representativeness; Timeliness, Other considerations in workload selection. Work load characterization Techniques: Terminology; Averaging, Specifying dispersion, Single Parameter Histograms, Multi Parameter Histograms, Principle Component Analysis, Markov Models, Clustering.

10 Hours

MODULE III

Monitors, Program Execution Monitors and Accounting Logs: Monitors: Terminology and classification; Software and hardware monitors, Software versus hardware monitors, Firmware and hybrid monitors, Distributed System Monitors, Program Execution Monitors and Accounting Logs, Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and Interpretation of Accounting log data, Using accounting logs to answer commonly asked questions.

10 Hours

MODULE IV

Capacity Planning and Benchmarking: Steps in capacity planning and management; Problems in Capacity Planning; Common Mistakes in Benchmarking; Benchmarking Games; Load Drivers; Remote- Terminal Emulation; Components of an RTE; Limitations of RTEs. **Experimental Design and Analysis: Introduction**: Terminology, Common mistakes in experiments, Types of experimental designs, 2k Factorial Designs, Concepts, Computation of effects, Sign table method for computing effects; Allocation of variance; General 2k Factorial Designs, General full factorial designs with k factors: Model, Analysis of a General Design, Informal Methods.

10 Hours

MODULE V

Queuing Models: Introduction: Queuing Notation; Rules for all Queues; Little's Law, Types of Stochastic Process. Analysis of Single Queue: Birth-Death Processes; M/M/1 Queue; M/M/m Queue; M/M/m/B Queue with finite buffers; Results for other M/M/1 Queuing Systems. Queuing Networks: Open and Closed Queuing Networks; Product form networks, queuing Network models of Computer Systems. Operational Laws: Utilization Law; Forced Flow Law; Little's Law; General Response Time Law; Interactive Response Time Law; Bottleneck Analysis; Mean Value Analysis and Related Techniques; Analysis of Open Queuing Networks; Mean Value Analysis; Approximate MVA; Balanced Job Bounds; Convolution Algorithm, Distribution of Jobs in a System, Convolution Algorithm for Computing G(N), Computing Performance using G(N), Timesharing Systems, Hierarchical Decomposition of Large Queuing Networks: Load Dependent Service Centers, Hierarchical Decomposition, Limitations of Queuing Theory. **10 Hours**

COURSE OUTCOMES:

The students shall be able to:

- Identify the need for performance evaluation and the metrics used for it
- Implement Little's law and other operational laws
- Apply the operational laws to open and closed systems
- Use discrete-time and continuous-time Markov chains to model real world systems
- Develop analytical techniques for evaluating scheduling policies

TEXT BOOK:

1. Raj Jain: The Art of Computer Systems Performance Analysis, John Wiley and Sons, 2013.

REFERENCE BOOKS:

1. Paul J Fortier, Howard E Michel: computer Systems Performance Evaluation and prediction, Elsevier, 2003.
2. Trivedi K S: Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd Edition, Wiley India, 2001.

SEM I **Year 2014-15**

Course Title: Distributed Operating System	**Course Code: 14SCE152**
Credits(L:T:P): 4:0:0	**Core/Elective: Elective**
Type Of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVE:

- To explore distributed systems principles associated with communication, naming, synchronization, distributed file systems, system design, distributed scheduling, and several case studies
- To cover both foundational concepts and well as practical deployments.
- To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- To gain insight on to the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols

TOPICS:

MODULE I

Fundamentals: What is Distributed Computing Systems? Evolution of Distributed Computing System; Distributed Computing System Models; What is Distributed Operating System? Issues in Designing a Distributed Operating System; Introduction to Distributed Computing Environment (DCE). **Message Passing:** Introduction, Desirable features of a Good Message Passing System, Issues in PC by Message Passing, Synchronization, Buffering, Multi-datagram Messages, Encoding and Decoding of Message Data, Process Addressing, Failure Handling, Group Communication, Case Study: 4.3 BSD UNIX IPC Mechanism.

10 Hours

MODULE II

Remote Procedure Calls: Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC.

10 Hours

MODULE III

Distributed Shared Memory: Introduction, General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM. **Synchronization:** Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms.

10 Hours

MODULE IV

Resource Management: Introduction, Desirable Features of a Good Global Scheduling Algorithm, Task Assignment Approach, Load – Balancing Approach, Load – Sharing Approach **Process Management:** Introduction, Process Migration, Threads. **10 Hours**

MODULE V

Distributed File Systems: Introduction, Desirable Features of a Good Distributed File System, File models, File– Accessing Models, File – Sharing Semantics, File – Caching Schemes, File Replication, Fault Tolerance, Atomic Transactions and Design Principles. **10 Hours**

COURSE OUTCOMES:

The students shall be able to:

- The concepts underlying distributed systems
- Demonstrate an ability to apply theory and techniques to unseen problems.
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Explore the various resource management techniques for distributed systems

TEXT BOOK:

1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.

REFERENCE BOOK:

1 Andrew S. Tanenbaum: Distributed Operating Systems, Pearson Education, 2013.

SEM I **Year 2014-15**

Course Title: Software Agents	**Course Code: 14SCE153**
Credits(L:T:P): 4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To learn the principles and fundamentals of designing agents
- To study the architecture design of different agents.
- To learn to do detailed design of the agents
- To understand user interaction with agents
- To explore the role of agents in assisting the users in day to day activities

TOPICS:

MODULE I

An introduction to Software Agents

Why Software Agents? Simplifying Computing, Barriers to Intelligent Interoperability, Incorporating Agents as Resource Managers, Overcoming user Interface Problems, Toward Agent-Enabled System Architectures. **Agents: From Direct Manipulation to Delegation** Introduction, Intelligent Interfaces, Digital Butlers, Personal Filters, Digital sisters-in-Law, Artificial Intelligence, Decentralization, Why Linking works, The Theatrical Metaphor, Conclusion: Direct Manipulation and Digital Butlers, Acknowledgements. **Interfaces Agents Metaphors with Character** Introduction, Objections to Agents, In Defense of Anthropomorphism, Key Characteristics of Interface Agents, Agency, Responsiveness, Competence, Accessibility, Design and Dramatic Character, An R & D Agenda **10 Hours**

MODULE II

Designing Agents as if People Mattered: What does "Agents" Mean?, Adaptive Functionality: Three Design Issues, The Agent Metaphor: Reactions and Expectations The Agent Conceptual Model. **Direct Manipulation versus Agents: Paths to Predictable, Controllable, and Comprehensible Interfaces:** Introduction, General Concerns About Intelligent Interfaces, Learning From History, What Is an Agent?, Looking at the Components, Realizing a New Vision, Tree Maps, Dynamic Queries, Back to a Scientific Approach, Acknowledgements. **Agents for Information Sharing and Coordination: A History and some Reflections:** Information, Lens: An Intelligent Tool for Managing Electronic Messages, Semiformal Systems and Radical Tailorability, Oval: A Radically Tailorable Tool for Information Management and Cooperative Work, Examples of Application and Agents in Oval, Conclusions: An Addendum: The Relationship between Oval and Objects Lens

10 Hours

MODULE III

Agents that Reduce Work and Information Overload Introduction, Approaches to Building Agents, Training a Personal Digital Assistant, Some Example of Existing Agents, Electronic Mail Agents, Meeting Scheduling Agent, News Filtering Agent, Entertainment Selection Agent, Discussion, Acknowledgements **Software Agents for Cooperative Learning:** Computer-Supported Cooperative Learning, Examples of Software Agents for Cooperative Learning, Examples of Software Agents for Cooperative Learning, Developing an Example, Discussion and Perspectives.

10 Hours

MODULE IV

An Overview of Agent-Oriented Programming: Agent-Oriented Programming: Software with Mental State, Two Scenarios, On the Mental state of agents, Generic Agent Interpreter, AGENT-0: A Simple Language and its Interpreter, **KQML as an Agent Communication Language:** The approach of knowledge sharing effort(KSE), The Solution of the knowledge sharing efforts, knowledge Query Manipulation Language (KQML),Implementation, Application of KQML , Other Communication Language, The Approach of Knowledge-Sharing Effect,(KSE),The Solutions of the Sharing Effect,

10 Hours

MODULE V

Agent for Information Gathering: Agent Organization, The Knowledge of an Agent, The Domain Model of an Agent, Modeling other Agent, communication language and protocol, query processing, an information goal, information source selection, generating a query access plan, interleaving planning and execution , semantic query optimization, learning, caching retrieved data, related work, discursion, acknowledgement. **Mobile Agents:** Enabling Mobile Agents, Programming Mobile Agents, Using Mobile Agents

10 Hours

COURSE OUTCOMES:

The student shall be able to:

- Identify and explore the advantages of agents
- Design the architecture for an agent
- Design the agent in details in a view for the implementation
- Design communicative actions with agents.

- Design typical agents using a tool for different types of applications.

TEXT BOOK:

1. Jeffrey M. Bradshaw: Software Agents, PHI (MIT Press) 2012.

REFERENCES:

1. Lin Padgham and Michael Winikoff, "Developing Intelligent Agent Systems: A Practical Guide", John Wiley & sons Publication, 2004.
2. Steven F. RailsBack and Volker Grimm, "Agent-Based and Individual Based modeling: A Practical Introduction", Princeton University Press, 2012.

SEM I **Year 2014-15**

Course Title: Bio-Informatics	**Course Code: 14SCE154**
Credits(L:T:P): 4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To get exposed to the domain of bioinformatics
- To understand the role of data warehousing and data mining for bioinformatics
- To learn to model bioinformatics based applications
- To understand how to deploy the pattern matching and visualization techniques in bioinformatics
- To study the Microarray technologies for genome expression

TOPICS:

MODULE I

INTRODUCTION : Need for Bioinformatics technologies – Overview of Bioinformatics technologies – Structural bioinformatics – Data format and processing – secondary resources- Applications – Role of Structural bioinformatics - Biological Data Integration System.

10 Hours

MODULE II

DATAWAREHOUSING AND DATAMINING IN BIOINFORMATICS: Bioinformatics data – Data ware housing architecture – data quality – Biomedical data analysis – DNA data analysis – Protein data analysis – Machine learning – Neural network architecture- Applications in bioinformatics.

10 Hours

MODULE III

MODELING FOR BIOINFORMATICS : Hidden markov modeling for biological data analysis – Sequence identification – Sequence classification – multiple alignment generation – Comparative modeling – Protein modeling – genomic modeling – Probabilistic modeling – Bayesian networks – Boolean networks - Molecular modeling – Computer programs for molecular modeling. **10 Hours**

MODULE IV

PATTERN MATCHING AND VISUALIZATION: Gene regulation – motif recognition and motif detection – strategies for motif detection – Visualization – Fractal analysis – DNA walk models – one dimension – two dimension – higher dimension – Game representation of Biological sequences – DNA, Protein, Amino acid sequences. **10 Hours**

MODULE V

MICROARRAY ANALYSIS: Microarray technology for genome expression study – image analysis for data extraction – preprocessing – segmentation – gridding, spot extraction, normalization, filtering – cluster analysis – gene network analysis. **10 Hours**

COURSE OUTCOMES

The students shall able to:

- Deploy the data warehousing and data mining techniques in Bioinformatics
- Model bioinformatics based applications
- Deploy the pattern matching and visualization techniques in bioinformatics
- Work on the protein sequences
- Use the Microarray technologies for genome expression

TEXT BOOK:

1. Yi-Ping Phoebe Chen (Ed), "Bio Informatics Technologies", Springer Verlag, 2014.

REFERENCES:

1. Andreas D. Baxevanis, B.F. Francis Ouellette: Bio Informatics A Practical Guide to Analysis of Genes and Proteins, Willey India 2009.

SEM I **Year 2014-15**

Course Title: Advanced Digital Design Laboratory	**Course Code: 14SCE16**
Credits(02) (L:T:P): 0:0:3	**Core/Elective: Core**
Type of Course: Practical	**Total Contact Hours:42**

COURSE OBJECTIVES

- To learn about various IC technology options
- To conduct simulation and Design verification using Verilog.
- To implement Behavioral modeling, Boolean-Equation, Flip-Flops and Latches; multiplexers, encoders, and decoders, synchronizers for asynchronous signals.
- To implement combinational logic; registered logic; registers and counters; Resets; Divide and conquer: Partitioning a design.

LABORATORY WORK:

Note: Use appropriate tools/language to implement the following experiment:

1. Design, develop, and verify a Verilog module that implements a JK Edge-Triggered Flip-Flop with Active-Low Preset ad Clear Inputs.

2. Design, develop, and verify a Verilog module that produces a 4-bit output indicating the number of 1s in an 8-bit input word.

3. Design, develop, and verify a Verilog module that implements a Universal Shift Register and then implement a bidirectional ring counter capable of counting in either direction, beginning with first active clock edge after reset.

4. Design, develop, and verify a Verilog module that implements a counter whose modulus value $n \leq 10$.

5. Design, develop, and verify a Verilog module that implements a Hamming Encoder that produces a 7- bit Hamming code given a 4-bit input word.

6. Design, develop, and verify a Verilog module that implements a 16 bit cyclic redundancy check (CRC-16) algorithm. Which shall encode 14 bits of message with a 3-bit CRC.

Note: Student can verify the verilog module output using Xilinx or equivalent simulator and FPGA Kit

Course Outcomes:

The students shall be able to:

- Work on various IC technology options.
- Implement Logic simulation, Design verification, Verilog.
- Work on Flip-Flops and Latches; multiplexers, encoders, and decoders, synchronizers for asynchronous signals.
- Design and implement circuits on combinational logic; Registered logic; registers and counters; Resets; Divide and conquer: Partitioning a design.

SEM II **Year 2014-15**

Course Title: Managing Big Data	**Course Code: 14SCE21**
Credits(L:T:P):3:0:1	**Core/Elective: core**
Type of Course: Lecture & Practical	**Total Contact Hours:50**

COURSE OBJECTIVES

- To Understand big data for business intelligence
- To Learn business case studies for big data analytics
- To manage Big data Without SQL
- To understand map-reduce analytics using Hadoop and related tools
- To Explore more on Hadoop and related tools

TOPICS:

MODULE I

UNDERSTANDING BIG DATA

What is big data – why big data –.Data!, Data Storage and Analysis, Comparison with Other Systems, Rational Database Management System , Grid Computing, Volunteer Computing, convergence of key trends – unstructured data – industry examples of big data – web analytics – big data and marketing – fraud and big data – risk and big data – credit risk management – big data and algorithmic trading – big data and healthcare – big data in medicine – advertising and big data – big data technologies – introduction to Hadoop – open source

technologies – cloud and big data – mobile business intelligence – Crowd sourcing analytics – inter and trans firewall analytics. **10 Hours**

MODULE II

NOSQL DATA MANAGEMENT

Introduction to NoSQL – aggregate data models – aggregates – key-value and document data models – relationships – graph databases – schema less databases – materialized views – distribution models – shading — version – map reduce – partitioning and combining – composing map-reduce calculations.

10 Hours

MODULE III

BASICS OF HADOOP

Data format – analyzing data with Hadoop – scaling out – Hadoop streaming – Hadoop pipes – design of Hadoop distributed file system (HDFS) – HDFS concepts – Java interface – data flow – Hadoop I/O – data integrity – compression – serialization – Avro – file-based data structures. **10 Hours**

MODULE IV

MAPREDUCE APPLICATIONS

MapReduce workflows – unit tests with MRUnit – test data and local tests – anatomy of MapReduce job run – classic Map-reduce – YARN – failures in classic Map-reduce and YARN – job scheduling – shuffle and sort – task execution – MapReduce types – input formats – output formats **10 Hours**

MODULE V

HADOOP RELATED TOOLS

Hbase – data model and implementations – Hbase clients – Hbase examples –praxis. Cassandra – Cassandra data model – Cassandra examples – Cassandra clients –Hadoop integration. Pig – Grunt – pig data model – Pig Latin – developing and testing Pig Latin scripts. Hive – data types and file formats – HiveQL data definition – HiveQL data manipulation – HiveQL queries. **10 Hours**

LAB EXPERIMENTS

Exercise 1 --- HDFS

Start by reviewing HDFS. You will find that its composition is similar to your local Linux file system. You will use the hadoop fs command when interacting with HDFS.

1. Review the commands available for the Hadoop Distributed File System:
2. Copy file foo.txt from local disk to the user's directory in HDFS
3. Get a directory listing of the user's home directory in HDFS
4. Get a directory listing of the HDFS root directory
5. Display the contents of the HDFS file user/Fred/bar.txt
6. Move that file to the local disk, named as baz.txt
7. Create a directory called input under the user's home directory
8. Delete the directory input old and all its contents
9. Verify the copy by listing the directory contents in HDFS:

Exercise 2 --- MapReduce

1. Create a JOB and submit to cluster
2. Track the job information
3. Terminate the job

4. Counters in MR Jobs with example
5. Map only Jobs and generic map examples
6. Distributed cache example
7. Combiners, Secondary sorting and Job chain examples

Exercise 3 --- MapReduce (Programs)
Using movie lens data

1. List all the movies and the number of ratings
2. List all the users and the number of ratings they have done for a movie
3. List all the Movie IDs which have been rated (Movie Id with at least one user rating it)
4. List all the Users who have rated the movies (Users who have rated at least one movie)
5. List of all the User with the max, min, average ratings they have given against any movie
6. List all the Movies with the max, min, average ratings given by any user

Exercise4 – Extract facts using Hive
Hive allows for the manipulation of data in HDFS using a variant of SQL. This makes it excellent for transforming and consolidating data for load into a relational database. In this exercise you will use HiveQL to filter and aggregate click data to build facts about user's movie preferences. The query results will be saved in a staging table used to populate the Oracle Database.

The moveapp_log_json table contains an activity column. Activity states are as follows:
1. RATE_MOVIE
2. COMPLETED_MOVIE
3. PAUSE_MOVIE
4. START_MOVIE
5. BROWSE_MOVIE
6. LIST_MOVIE
7. SEARCH_MOVIE
8. LOGIN
9. LOGOUT
10. INCOMPLETE_MOVIE

```
hive> SELECT * FROM movieapp_log_json LIMIT 5;
hive> drop table movieapp_log_json;
hive> CREATE EXTERNAL TABLE movieapp_log_json (
custId INT,
movieId INT,
genreId INT,
time STRING,
recommended STRING,
activity INT,
rating INT,
price FLOAT
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.JsonSerde'
LOCATION '/user/oracle/moviework/applog/';
```

hive> SELECT * FROM movieapp_log_json LIMIT 20;

hive> SELECT MIN(time), MAX(time) FROM movieapp_log_json

1. PURCHASE_MOVIE
Hive maps queries into MapReduce jobs, simplifying the process of querying large datasets in HDFS. HiveQL statements can be mapped to phases of the MapReduce framework. As illustrated in the following figure, selection and transformation operations occur in map tasks, while aggregation is handled by reducers. Join operations are flexible: they can be performed in the reducer or mappers depending on the size of the leftmost table.
1. Write a query to select only those clicks which correspond to starting, browsing, completing, or purchasing movies. Use a CASE statement to transform the RECOMMENDED column into integers where 'Y' is 1 and 'N' is 0. Also, ensure GENREID is not null. Only include the first 25 rows.

2. Write a query to select the customer ID, movie ID, recommended state and most recent rating for each movie.

3. Load the results of the previous two queries into a staging table. First, create the staging table:

4. Next, load the results of the queries into the staging table.

Exercise 5 Extract sessions using Pig

While the SQL semantics of HiveQL are useful for aggregation and projection, some analysis is better described as the flow of data through a series of sequential operations. For these situations, Pig Latin provides a convenient way of implementing dataflow over data stored in HDFS. Pig Latin statements are translated into a sequence of MapReduce jobs on the execution of any STORE or DUMP command. Job construction is optimized to exploit as much parallelism as possible, and much like Hive, temporary storage is used to hold intermediate results. As with Hive, aggregation occurs largely in the reduce
tasks. Map tasks handle Pig's FOREACH and LOAD, and GENERATE statements. The EXPLAIN command will show the execution plan for any Pig Latin script. As of Pig 0.10, the ILLUSTRATE command will provide sample results for each stage of the execution plan.

In this exercise you will learn basic Pig Latin semantics and about the fundamental types in Pig Latin, Data Bags and Tuples.

1. Start the Grunt shell and execute the following statements to set up a dataflow with the clickstream data. Note: Pig Latin statements are assembled into MapReduce jobs which are launched at execution of a DUMP or STORE statement.

2. Group the log_sample by movie and dump the resulting bag.

3. Add a GROUP BY statement to the sessionize.pig script to process the clickstream data into user sessions.

COURSE OUTCOMES:
The students shall able to:

- Describe big data and use cases from selected business domains
- Explain NoSQL big data management
- Install, configure, and run Hadoop and HDFS
- Perform map-reduce analytics using Hadoop
- Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data Analytics

TEXT BOOKS:

1. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
2. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.

REFERENCES:

1.Vignesh Prajapati, Big data analytics with R and Hadoop, SPD 2013.
2. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
3. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.
4. Alan Gates, "Programming Pig", O'Reilley, 2011

SEM II **Year 2014-15**

Course Title: Mobile Application Development	**Course Code: 14SCE22**
Credits(L:T:P):3:0:1	**Core/Elective: core**
Type of Course: Lecture & Practical	**Total Contact Hours:50**

COURSE OBJECTIVES

- To Understand system requirements for mobile applications
- To learn basics mobile development frameworks
- To deign mobile applications
- To learn and implement mobile applications

TOPICS:

MODULE I

Introduction to mobile communication and computing, Introduction to mobile computing, Novel applications, limitations and GSM architecture, Mobile services, System architecture, Radio interface, protocols, Handover and security. Smart phone operating systems and smart phones applications.

10 Hours

MODULE II

Fundamentals of Android Development: Introduction to Android., The Android 4.1 Jelly Bean SDK, Understanding the Android Software Stack, Installing the Android SDK, Creating Android Virtual Devices, Creating the First Android Project, Using the Text View Control, Using the Android Emulator, The Android Debug Bridge (ADB), Basic Widgets Understanding the Role of Android Application Components, Event Handling , Displaying Messages Through Toast, Creating and Starting an Activity, Using the Edit ext Control .

10 Hours

MODULE III

The Android Debug Bridge (ADB), Basic Widgets Understanding the Role of Android Application Components, Event Handling , Displaying Messages Through Toast, Creating and Starting an Activity, Using the Edit ext Control Building Blocks for Android Application Design, Laying Out Controls in Containers, Utilizing Resources and Media, Using Selection Widgets and Debugging Displaying and Fetching Information Using Dialogs and Fragments **10 Hours**

MODULE IV

Widgets and Debugging Using Selection Widgets and Debugging Displaying and Fetching Information Using Dialogs and Fragments Advanced Android Programming: Internet, Entertainment, and Services, Implementing dra wing and animations, **10 Hours**

MODULE V

Displaying web pages and maps: Displaying web pages and maps communicating with sms and emails,. creating and using content providers: Creating and consuming services, Publishing android applications.

10 Hours

LAB EXPERIMENTS:

Using Wireless Markup language develop the APP using Android OS

1 .Design and develop an Mobile App for smart phones The Easy Unit Converter using Android.
This application should have approximately 20 categories to be used in your daily life. It includes following units: Acceleration, Angle, Area, Circle, Capacitor , Cooking, Data Size, Density, Data Transfer rate, Electric Current, Energy,- Flow Rate , Force

2. Design and develop an Mobile App for smart phones Currency Converter.
This applications should synchronize online as you run it and sends you back the latest and most reliable exchange rates possible.

This application should support following conversions:

EUR->Euro	
GBP->British	Pound
USD->UnitedStates	Dollar
AUD->Australian	Dollar
CAD->Canadian	Dollar
CHF->Swiss	Franc
CNY->Chinese	Yuan
HKD->HongKong	Dollar
IDR->Indonesian	Rupiah
INR->Indian	Rupee
JPY->Japanese	Yen
THB->Thai	Baht

3. Design and develop an Mobile App game for smart phones The Tic Tac Toe using Android.

4 Design and develop an Mobile App for smart phones ,The Health Monitoring System using Android. This App should record Biochemistry Lab Parameters and if abnormal should send an SMS to doctor for Medications.

5 Design and develop an Mobile App for smart phones The Expense Manager using Android.
This is an application for managing your expenses and incomes: Tracking expenses and incomes by week, month and year as well as by categories, Multiple accounts in multiple currencies, Schedule the payments and recurring payments, Take a picture of receipt, Payment alerts, Budget by day, week, month and year, Search and reports, Import and export account activities in CSV for desktop software, Customize expense categories, payer/payer, payment methods, date format, white or black background, button style etc, Account transfer, Convenient tools such calculator, currency converter, tip calculator, sales and tax calculator and credit card calculator.

COURSE OUTCOMES:

The students shall able to:

- Describe the requirements for mobile applications
- Explain the challenges in mobile application design and development
- Develop design for mobile applications for specific requirements
- Implement the design using Android SDK
- Implement the design using Objective C and iOS
- Deploy mobile applications in Android and iPone marketplace for distribution

TEXT BOOKS:

1. Mobile Computing: technologies and Applications- N. N. Jani S chand2009.

2. B.M.Hirwani- Android programming Pearson publications-2013

SEM II **Year 2014-15**

Course Title: Wireless Networks And Mobile Computing	**Course Code: 14SCE23**
Credits(L:T:P):4:0:0	**Core/Elective: Core**
Type Of Course: Lecture	**Total Contact Hours:50**

COURSE OUTCOMES

- To introduce the concepts of wireless communication.
- To understand various propagation methods, Channel models, capacity calculations multiple antennas and multiple user techniques used in the mobile communication.
- To understand CDMA, GSM. Mobile IP, WImax
- To understand Different Mobile OS
- To learn various Markup Languages
- CDC, CLDC, MIDP; Programming for CLDC, MIDlet model and security concerns

TOPICS:

MODULE I

Mobile Computing Architecture: Architecture for Mobile Computing, 3-tier Architecture, Design Considerations for Mobile Computing. **Wireless Networks** : Global Systems for Mobile Communication (GSM and Short Service Messages (SMS): GSM Architecture, Entities, Call routing in GSM, PLMN Interface, GSM Addresses and Identities, Network Aspects in GSM, Mobility Management, GSM Frequency allocation. Introduction to SMS, SMS Architecture, SM MT, SM MO, SMS as Information bearer, applications, GPRS and Packet Data Network, GPRS Network Architecture, GPRS Network Operations, Data Services in GPRS, Applications for GPRS, Billing and Charging in GPRS, Spread Spectrum technology, IS-95, CDMA versus GSM, Wireless Data, Third Generation Networks, Applications on 3G, Introduction to WiMAX.

10 Hours

MODULE II

Mobile Client: Moving beyond desktop, Mobile handset overview, Mobile phones and their features, PDA, Design Constraints in applications for handheld devices. Mobile IP: Introduction, discovery, Registration, Tunneling, Cellular IP, Mobile IP with IPv6

10 Hours

MODULE III

Mobile OS and Computing Environment : Smart Client Architecture, The Client: User Interface, Data Storage, Performance, Data Synchronization, Messaging. The Server: Data Synchronization, Enterprise Data Source, Messaging. Mobile Operating Systems: WinCE, Palm OS, Symbian OS, Linux, Proprietary OS Client Development: The development process, Need analysis phase, Design phase, Implementation and Testing phase, Deployment phase, Development Tools, Device Emulators. **10 Hours**

MODULE IV

Building, Mobile Internet Applications: Thin client: Architecture, the client, Middleware, messaging Servers, Processing a Wireless request, Wireless Applications Protocol (WAP) Overview, Wireless Languages: Markup Languages, HDML, WML, HTML, cHTML, XHTML, VoiceXML. **10 Hours**

MODULE V

J2ME: Introduction, CDC, CLDC, MIDP; Programming for CLDC, MIDlet model, Provisioning, MIDlet life-cycle, Creating new application, MIDlet event handling, GUI in MIDP, Low level GUI Components, Multimedia APIs; Communication in MIDP, Security Considerations in MIDP. **10 Hours**

COURSE OUTCOMES:

The students shall able to:

- Work on state of art techniques in wireless communication.
- Explore CDMA, GSM. Mobile IP, WImax
- Work on Different Mobile OS
- Develop program for CLDC, MIDP let model and security concerns

TEXT BOOKS:

1. Ashok Talukder, Roopa Yavagal, Hasan Ahmed: Mobile Computing, Technology, Applications and Service Creation, 2nd Edition, Tata McGraw Hill, 2010.

2. Martyn Mallik: Mobile and Wireless Design Essentials, Wiley India, 2003

REFERENCE BOOKS:

1. Raj kamal: Mobile Computing, Oxford University Press, 2007.

2. Iti Saha Misra: Wireless Communications and Networks, 3G and Beyond, Tata McGraw Hill, 2009.

SEM II **Year 2014-15**

Course Title: Multi-Core Architecture And Programming	**Course Code: 14SCE24**
Credits(L:T:P):4:0:0	**Core/Elective: core**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES:

- To understand the recent trends in the field of Computer Architecture and identify performance related parameters
- To appreciate the need for parallel processing
- To expose the students to the problems related to multiprocessing
- To understand the different types of multicore architectures
- To understand concepts of multi threading, OPENMP.

TOPICS:

MODULE I

Introduction to Multi-core Architecture Motivation for Concurrency in software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-core Architectures from Hyper-Threading Technology, Multi-threading on Single-Core versus Multi-Core Platforms Understanding Performance, Amdahl's Law, Growing Returns: Gustafson's Law. **System Overview of Threading** : Defining Threads, System View of Threads, Threading above the Operating System, Threads inside the OS, Threads inside the Hardware, What Happens When a Thread Is Created, Application Programming Models and Threading, Virtual Environment: VMs and Platforms, Runtime Virtualization, System Virtualization.

10 Hours

MODULE II

Fundamental Concepts of Parallel Programming : Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Challenges You'll Face, Parallel Programming Patterns, A Motivating Problem: Error Diffusion, Analysis of the Error Diffusion Algorithm, An Alternate Approach: Parallel Error Diffusion, Other Alternatives. **Threading and Parallel Programming Constructs:** Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control- based Concepts, Fence, Barrier, Implementation-dependent Threading Features. **10 Hours**

MODULE III

Threading APIs : Threading APIs for Microsoft Windows, Win32/MFC Thread APIs, Threading APIs for Microsoft. NET Framework, Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads, Managing Threads, Thread Synchronization, Signaling, Compilation and Linking. **10 Hours**

MODULE IV

OpenMP: A Portable Solution for Threading : Challenges in Threading a Loop, Loop-carried Dependence, Data-race Conditions, Managing Shared and Private Data, Loop Scheduling and Portioning, Effective Use of Reductions, Minimizing Threading Overhead, Work-sharing Sections, Performance-oriented Programming, Using Barrier and No wait, Interleaving Single-thread and Multi-thread Execution, Data Copy-in and Copy-out, Protecting Updates of Shared Variables, Intel Task queuing Extension to OpenMP, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, performance.

10 Hours

MODULE V

Solutions to Common Parallel Programming Problems : Too Many Threads, Data Races, Deadlocks, and Live Locks, Deadlock, Heavily Contended Locks, Priority Inversion, Solutions for Heavily Contended Locks, Non-blocking Algorithms, ABA Problem, Cache Line Ping-ponging, Memory Reclamation Problem, Recommendations, Thread-safe Functions and Libraries, Memory Issues, Bandwidth, Working in the Cache, Memory Contention, Cache-related Issues, False Sharing, Memory Consistency, Current IA-32 Architecture, Itanium Architecture, High-level Languages, Avoiding Pipeline Stalls on IA-32,Data Organization for High Performance. **10 Hours**

COURSE OUTCOMES:

The students shall able to:

- Identify the limitations of ILP and the need for multicore architectures
- Solve the issues related to multiprocessing and suggest solutions
- Point out the salient features of different multicore architectures and how they exploit parallelism

TEXT BOOK:

1. Multicore Programming , Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts , Intel Press , 2006

SEM II **Year 2014-15**

Course Title: Data Mining & Data Warehousing	**Course Code: 14SCE251**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To expose the students to the concepts of Data warehousing Architecture and Implementation
- To Understand Data mining principles and techniques and Introduce DM as a cutting edge business intelligence
- To learn to use association rule mining for handling large data
- To understand the concept of classification for the retrieval purposes
- To know the clustering techniques in details for better organization and retrieval of data

TOPICS:

MODULE I

Introduction and Data Preprocessing : Why data mining, What is data mining, What kinds of data can be mined, What kinds of patterns can be mined, Which Technologies Are used, Which kinds of Applications are targeted, Major issues in data mining .Data Preprocessing: An overview, Data cleaning, Data integration, Data reduction, Data transformation and data discretization. **10 Hours**

MODULE II

Data warehousing and online analytical processing: Data warehousing: Basic concepts, Data warehouse modeling: Data cube and OLAP, Data warehouse design and usage, Data warehouse implementation, Data generalization by attribute-oriented induction, **10 Hours**

MODULE III

Classification: Basic Concepts: Basic Concepts, Decision tree induction, Bays Classification Methods, Rule-Based classification, Model evaluation and selection, Techniques to improve classification accuracy.

10 Hours

MODULE IV

Cluster Analysis: Basic concepts and methods: Cluster Analysis, Partitioning methods, Hierarchical Methods, Density-based methods, Grid-Based Methods, Evaluation of clustering. **10 Hours**

MODULE V

Data mining trends and research frontiers: Mining complex data types, other methodologies of data mining, Data mining applications, Data Mining and society. **10 Hours**

COURSE OUTCOMES:

The students shall able to:

- Store voluminous data for online processing
- Preprocess the data for mining applications
- Apply the association rules for mining the data
- Design and deploy appropriate classification techniques
- Cluster the high dimensional data for better organization of the data
- Discover the knowledge imbibed in the high dimensional system

TEXT BOOK:

1. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining Concepts and Techniques, ELSEVIER(MK) 3rd edition 2012.

SEM II **Year 2014-15**

Course Title: Pattern Recognition	**Course Code: 14SCE252**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

Course Objectives:

- To introduce the student to various Image processing and Pattern recognition techniques.
- To study the mathematical morphology necessary for Pattern recognition.
- To study the Image Representation and description and feature extraction.
- To study the principles of decision trees and clustering in pattern recognition.

TOPICS:

MODULE I

Introduction: Definition of PR, Applications, Datasets for PR, Different paradigms for PR, Introduction to probability, events, random variables, Joint distributions and densities, moments. Estimation minimum risk estimators, problems **10 Hours**

MODULE II

Representation: Data structures for PR, Representation of clusters, proximity measures, size of patterns, Abstraction of Data set, Feature extraction, Feature selection, Evaluation **10 Hours**

MODULE III

Nearest Neighbor based classifiers & Bayes classifier: Nearest neighbor algorithm, variants of NN algorithms, use of NN for transaction databases, efficient algorithms, Data reduction, prototype selection, Bayes theorem, minimum error rate classifier, estimation of probabilities, estimation of probabilities, comparison with NNC, Naive bayes classifier, Bayessian belief network **10 Hours**

MODULE IV

Decision Trees: Introduction, DT for PR, Construction of DT, Splitting at the nodes, Over fitting & Pruning, Examples **10 Hours**

MODULE V

Clustering: Hierarchical (Agglomerative, single/complete/average linkage, wards, Partitional (Forgy's, k-means, Isodata), clustering large data sets, examples **10 Hours**

COURSE OUTCOMES:

The students shall able to:

- Develop algorithms for Pattern Recognition.
- Develop and analyze decision tress.
- Design the nearest neighbor classifier.

Text Book:

1. Pattern Recognition (An Introduction) , V Susheela Devi, M Narsimha Murthy, 2011 Universities Press, ISBN 978-81-7371-725-3
2. Pattern Recognition & Image Analysis, Earl Gose, Richard Johnsonbaugh, Steve Jost. PHI ISBN-81-203-1484-0, 1996.

References

1. Duda R. O., P.E. Hart, D.G. Stork., Pattern Classification, John Wiley and sons, 2000.

SEM II **Year 2014-15**

Course Title: Advances In Storage Area Networks	**Course Code: 14SCE253**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- To understand the fundamentals of storage centric and server centric systems
- To understand the metrics used for Designing storage area networks
- To understand the RAID concepts
- To enable the students to understand how data centre's maintain the data with the concepts of backup mainly remote mirroring concepts for both simple and complex systems

TOPICS:

MODULE I

Introduction: Server Centric IT Architecture and its Limitations; Storage – Centric IT Architecture and its advantages. Case study: Replacing a server with Storage Networks The Data Storage and Data Access problem; The Battle for size and access. Intelligent Disk Subsystems: Architecture of Intelligent Disk Subsystems; Hard disks and Internal I/O Channels; JBOD, Storage virtualization using RAID and different RAID levels; Caching: Acceleration of Hard Disk Access; Intelligent disk subsystems, Availability of disk subsystems. **10 Hours**

MODULE II

I/O Techniques: The Physical I/O path from the CPU to the Storage System; SCSI; Fibre Channel Protocol Stack; Fibre Channel SAN; IP Storage. Network Attached Storage: The NAS Architecture, The NAS hardware Architecture, The NAS Software Architecture, Network connectivity, NAS as a storage system. File System and NAS: Local File Systems; Network file Systems and file servers; Shared Disk file systems; Comparison of fibre Channel and NAS.

10 Hours

MODULE III

Storage Virtualization: Definition of Storage virtualization; Implementation Considerations; Storage virtualization on Block or file level; Storage virtualization on various levels of the storage Network; Symmetric and Asymmetric storage virtualization in the Network. **10 Hours**

MODULE IV

SAN Architecture and Hardware devices: Overview, Creating a Network for storage; SAN Hardware devices; The fibre channel switch; Host Bus Adaptors; Putting the storage in SAN; Fabric operation from a Hardware perspective. Software Components of SAN: The switch's Operating system; Device Drivers; Supporting the switch's components; Configuration options for SANs.

10 Hours

MODULE V

Management of Storage Network: System Management, Requirement of management System, Support by Management System, Management Interface, Standardized Mechanisms, Property Mechanisms, In-band Management, Use of SNMP, CIM and WBEM, Storage Management Initiative Specification (SMI-S), CMIP and DMI, Optional Aspects of the Management of Storage Networks, Summary **10 Hours**

COURSE OUTCOMES:

The students shall able to:

- Identify the need for performance evaluation and the metrics used for it
- Apply the techniques used for data maintenance.
- Realize strong virtualization concepts
- Develop techniques for evaluating policies for LUN masking, file systems

Text Book:

1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, Wiley India, 2013.

Reference Books:

1. Robert Spalding: "Storage Networks The Complete Reference", Tata McGraw-Hill, 2011.
2. Marc Farley: Storage Networking Fundamentals – An Introduction to Storage Devices, Subsystems, Applications, Management, and File Systems, Cisco Press, 2005.
3. Richard Barker and Paul Massiglia: "Storage Area Network Essentials A Complete Guide to understanding and Implementing SANs", Wiley India, 2006.

SEM II **Year 2014-15**

Course Title: Decision Support Systems	**Course Code: 14SCE254**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- Understand the fundamentals of decision making and problem solving.
- Know the fundamentals of mathematical modeling.
- Know how to use an electronic spreadsheet as a mathematical model.
- Be familiar with how artificial intelligence emerged as a computer application, and its main areas.
- Know the four basic parts of an expert system.

- Know what a group decision support system (GDSS) is and the different environmental settings that can be used

TOPICS:

MODULE I

Introduction to decision support systems: DSS Defined, History of decision support systems, Ingredients of a DSS, Data and model management, DSS Knowledge base, User interfaces, User interfaces, The DSS user, Categories and classes of DSSs, Chapter Summary. **Decisions and decision makers** Decision makers: who are they, Decision styles, Decision effectiveness, How can a DSS help?, A Typology of decisions, Decision theory and simon's model of problem solving, Bounded decision making, The process of choice, Cognitive processes, Biases and heuristics in decision making, Chapter summary. **10 Hours**

MODULE II

Decisions in the organization: Understanding the organization, Organizational culture. **Modeling decision processes:** Defining the problem and its structures, Decision models, Types of probability, Techniques for forecasting probabilities, Calibration and sensitivity, Chapter summary.

10 Hours

MODULE III

Group decision support and groupware technologies: Group Decision making, the problem with groups, MDM support technologies, Managing MDM activities, the virtual workspace, chapter summary. **Executive information systems:** What exactly is an EIS, Some EIS history, Why area top executives so different?, EIS components, Making the EIS work, The future of executive decision making and the EIS, chapter summary. **10 Hours**

MODULE IV

Designing and building decision support systems: Strategies for DSS analysis and design, The DSS developer, DSS user interface issues, chapter summary. **Implementing and integrating decision support systems:** DSS implementation, System evaluation, The importance of integration, chapter summary. **10 Hours**

MODULE V

Creative decision making and problem solving What is creativity?, Creativity defined, The occurrence of creativity, Creative problem solving techniques, Creativity and the role of technology, chapter summary. **10 Hours**

COURSE OUTCOMES

The students shall able to:

- Recognize the relationship between business information needs and decision making
- Appraise the general nature and range of decision support systems
- Appraise issues related to the development of DSS
- Select appropriate modeling techniques
- Analyze, design and implement a DSS

TEXT BOOK:

1. George M.Marakas: DECISION SUPPORT SYSTEM, PHI.2011.

SEM II **Year 2014-15**

Course Title: Wireless Networks & Mobile Computing Laboratory	Course Code: 14SCE26
Credits(02) (L:T:P):0:0:3	Core/Elective: Core
Type of Course: Practical	Total Contact Hours:42

COURSE OUTCOMES

- To introduce the concepts of wireless communication.
- To implement propagation methods, Channel models, capacity calculations multiple antennas and multiple user techniques used in the mobile communication.
- To understand CDMA, GSM, Mobile IP, WiMax
- To work on different Mobile OS
- To implement various Markup Languages

LABORATORY WORK:

Note: Use appropriate tools/language to implement the following experiment:

1. Using any package like MATLAB or using any programming language of your choice, implement the BPSK algorithm and study its performance.

2. Using any package like MATLAB or using any programming language of your choice, implement the QPSK algorithm and study its performance.

3. Implement and study the performance of GSM on NS2(using MAC layer) or equivalent environment.

4. Implement and study the performance of CDMA on NS2(using stack called callnet) or equivalent environment.

5. Develop and create a currency convertor on **MIDlet** application with NetBeans or equivalent environment.

COURSE OUTCOMES:

The students shall able to:

- Work on state of art techniques in wireless communication.
- Hands on experience on Different Mobile OS
- Should be able to develop program for CLDC, MIDP let model and security concerns.

SEM IV **Year 2014-15**

Course Title: Arm Processors	**Course Code: 14SCE41**
Credits(L:T:P):3:0:1	**Core/Elective: Core**
Type of Course: Lecture & Practical	**Total Contact Hours:50**

COURSE OBJECTIVES

- Describe the programmer's model of ARM processor and create and test assembly level programming.
- Analyze various types of coprocessors and design suitable co-processor interface to ARM processor.
- Analyze floating point processor architecture and its architectural support for higher level language.
- Become aware of the Thumb mode of operation of ARM.
- Identify the architectural support of ARM for operating system and analyze the function of memory Management unit of ARM.

TOPICS:

MODULE I

An Introduction to Processor Design: Processor architecture and organization. Abstraction in hardware design. A simple processor. Instruction set design. Processor design trade-offs. The Reduced Instruction Set Computer. Design for low power consumption. **The ARM Architecture: T**he Acorn RISC Machine. Architectural inheritance. The ARM programmer's model. ARM development tools. **10 Hours**

MODULE II

ARM Assembly Language Programming: Data processing instructions. Data transfer instructions. Control flow instructions. Writing simple assembly language programs. **ARM Organization and Implementation:** 3-stage pipeline ARM organization. 5-stage pipeline ARM organization. ARM instruction execution. ARM implementation. The ARM coprocessor interface. **10 Hours**

MODULE III

The ARM Instruction Set: Introduction. Exceptions. Conditional execution. Branch and Branch with Link (B, BL) Branch, Branch with Link and exchange instructions (BX, BLX). Software Interrupt (SWI). Data processing instructions. Multiply instructions. Count leading zeros (CLZ - architecture v5T only). Single word and unsigned byte data transfer instructions. Half-word and signed byte data transfer instructions. Multiple register transfer instructions. Swap memory and register instructions (SWP). Status register to general register transfer instructions . General register to status register transfer instructions. Coprocessor instructions. Coprocessor data operations . Coprocessor data transfers. Coprocessor register transfers. Breakpoint instruction (BRK - architecture v5T only). Unused instruction space. Memory faults. ARM architecture variants. **Architectural Support for High-Level Languages:** Abstraction in software design. Data types. Floating-point data types. The ARM floating-point architecture. Expressions . Conditional statements. Loops. Functions and procedures. Use of memory. Run-time environment. **10 Hours**

MODULE IV

The Thumb Instruction Set: The Thumb bit in the CPSR .The Thumb programmer's model. Thumb branch instructions. Thumb software interrupts instruction. Thumb data processing instructions. Thumb single register data transfer instructions. Thumb multiple register data transfer instructions. Thumb breakpoint instruction. Thumb implementation. Thumb applications. **Architectural Support for System Development:** The ARM memory interface. The Advanced Microcontroller Bus Architecture (AMBA). The ARM reference peripheral specification. Hardware system prototyping tools. The JTAG boundary scan test architecture. The ARM debug architecture. Embedded Trace. Signal processing support. **10 Hours**

MODULE V

ARM Processor Cores: ARM7TDMI. ARM8. ARM9TDMI.ARM10TDMI **Memory Hierarchy:** Memory size and speed. On-chip memory. Memory management. **Architectural Support for Operating Systems.** An introduction to operating systems. The ARM system control coprocessor. CP15 protection unit registers. ARM protection unit. CP15 MMU registers. ARM MMU architecture. Synchronization. Context switching. Input / Output. **10 Hours**

ARM LABORATORY EXPERIMENTS

Carryout the following experiments using keil micro vision 4 software/ Equivalent tool.

1. Write a program to interface LCD to ARM kit.
2. Write a program to (a) copy a string from source to destination (b) Reverse a string.
3. Write a program to multiply two matrices with and without MLA instruction.
4. Write a program to scan the keypad, assign own values to the keys and display the key pressed.
5. Write a program to open a file and using fork system call create a child process. Let both the parent and child process write to the same file. Check the output of the file.
6. Write a program to communicate between two processes using (a) PIPE (b) FIFO.
7. Write a program to synchronize shared memory usage using Semaphore.
8. (a) Write a simple program to create three threads.
   (b) Perform 3x3 matrices addition using threads.

COURSE OUTCOMES:

The students shall able to:

- Understand the hardware and software issues related to the design of a Microcontroller based system catering to the needs of medium and higher end applications.
- Understand the architecture and programming of the 32-bit ARM Cortex Processors

Text Book:

1.Steve Furber:*ARM System on Chip Architecture* by S.B Fuber 2nd Edition, Pearson 2013.

Reference Book.

1. Joseph Yiu: The definitive guide to ARM Cortex M3 M4 processors, Elsevier Newnes 3rd edition 2014

SEM IV **Year 2014-15**

Course Title: Wireless Ad-hoc Networks	**Course Code: 14SCE421**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

Course Objective:

- To understand fundamental principles of Ad-hoc Networks
- To develop a comprehensive understanding of Ad-hoc network protocols
- To understand current and emerging trends in Ad-hoc Wireless Networks.
- To understand energy management in ad-hoc wireless networks.

TOPICS:

MODULE I

Ad-hoc Wireless Networks Introduction, Issues in Ad-hoc Wireless Networks, Ad-hoc Wireless Internet; **MAC Protocols for Ad-hoc Wireless Networks:** Introduction, Issues in Designing a MAC Protocol, Design Goals of MAC Protocols, Classification of MAC protocols, Contention-Based Protocols, Contention-Based Protocols with Reservation Mechanisms, Contention-Based Protocols with Scheduling Mechanisms, MAC Protocols that Use Directional Antennas (Chapter 5: 5.1-5.3, Chapter 6: 6.1-6.8)

10 Hours

MODULE II

Routing Protocols for Ad-hoc Wireless Networks Introduction, Issues in Designing a Routing Protocol for Ad-hoc Wireless Networks; Classification of Routing Protocols; Table Driven Routing Protocols; On-Demand Routing Protocols, Hybrid Routing Protocols, Hierarchical Routing Protocols and Power-Aware Routing Protocols (Chapter 7: 7.1-7.6, 7.8, 7.9) **10 Hours**

MODULE III

Multicast Routing in Ad-hoc Wireless Networks Introduction, Issues in Designing a Multicast Routing Protocol, Operation of Multicast Routing Protocols, An Architecture Reference Model for Multicast Routing Protocols, Classifications of Multicast Routing Protocols, Tree-Based Multicast Routing Protocols and Mesh-Based Multicast Routing Protocols. (Chapter 8: 8.1-8.7) **10 Hours**

MODULE IV

Transport Layer and Security Protocols for Ad-hoc Networks: Introduction, Issues in Designing a Transport Layer Protocol; Design Goals of a Transport Layer Protocol; Classification of Transport Layer Solutions; TCP over Transport Layer Solutions; Other Transport Layer Protocols for Ad-hoc Networks; Security in Ad-hoc Wireless Networks, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management and Secure Touting Ad-hoc Wireless Networks. (Chapter 9: 9.1-9.6, 9.7-9.12)

10 Hours

MODULE V

Quality of Service and Energy Management in Ad-hoc Wireless Networks: Introduction, Issues and Challenges in Providing QoS in Ad-hoc Wireless Networks, Classification of QoS Solutions, MAC Layer Solutions, Network Layer Solutions; Energy Management in Ad-hoc Wireless Networks: Introduction, Need for Energy Management in Ad-hoc Wireless Networks, Classification of Energy Management Schemes, Battery Management Schemes, Transmission Management Schemes, System Power Management Schemes. (Chapter 10: 10.1-10.5, Chapter 11: 11.1-11.6) **10 Hours**

Course Outcome:

The students shall able to:

- Design their own wireless network
- Evaluate the existing network and improve its quality of service

TEXT BOOKS:

1. C. Siva Ram Murthy & B. S. Manoj: Ad-hoc Wireless Networks, 2nd Edition, Pearson Education, 2011

REFERENCES:

1. Ozan K. Tonguz and Gianguigi Ferrari: Ad-hoc Wireless Networks, John Wiley, 2007.
2. Xiuzhen Cheng, Xiao Hung, Ding-Zhu Du: Ad-hoc Wireless Networking, Kluwer Academic Publishers, 2004.
3. C.K. Toh: Ad-hoc Mobile Wireless Networks- Protocols and Systems, Pearson Education, 2002.

SEM IV **Year 2014-15**

Course Title: Wireless Sensor Networks	**Course Code: 14SCE422**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

COURSE OBJECTIVES

- Architect sensor networks for various application setups.
- Explore the design space and conduct trade-off analysis between performance and resources.
- Assess coverage and conduct node deployment planning.
- Devise appropriate data dissemination protocols and model links cost.
- Determine suitable medium access protocols and radio hardware.
- Prototype sensor networks using commercial components.
- Provision quality of service, fault-tolerance, security and other dependability requirements while coping with resource constraints.

TOPICS:

MODULE I

Introduction, Overview and Applications of Wireless Sensor Networks

Introduction, Basic overview of the Technology, **Applications of Wireless Sensor Networks: I**ntroduction, Background, Range of Applications, Examples of Category 2 WSN Applications, Examples of Category 1 WSN Applications, Another Taxonomy of WSN Technology
(Chapter 1: 1.1, 1.2, Chapter2: 2.1-2.6) **10 Hours**

MODULE II

Basic Wireless Sensor Technology and Systems: Introduction, Sensor Node Technology, Sensor Taxonomy, WN Operating Environment, WN Trends, Wireless Transmission Technology and Systems: Introduction, Radio Technology Primer, Available Wireless Technologies (Chapter3: 3.1-3.5, Chapter 4: 4.1-4.3)

10 Hours

MODULE III

MAC and Routing Protocols for Wireless Sensor Networks:

Introduction, Background, Fundamentals of MAC Protocols, MAC Protocols for WSNs, Sensor-MAC case Study, IEEE 802.15.4 LR-WPANs Standard Case Study. **Routing Protocols for Wireless Sensor Networks:** Introduction, Background, Data Dissemination and Gathering, Routing Challenges and Design Issues in WSNs, Routing Strategies in WSNs. (Chapter 5: 5.1-5.6, Chapter 6: 6.1-6.5)

10 Hours

MODULE IV

Transport Control and Middleware for Wireless Sensor Networks: Traditional Transport Control Protocols, Transport Protocol Design Issues, Examples of Existing Transport Control Protocols, Performance of Transport

Control Protocols. **Middleware for Wireless Sensor Networks:** Introduction, WSN Middleware Principles, Middleware Architecture, Existing Middleware.
(Chapter 7: 7.1-7.4, Chapter 8: 8.1-8.4) **10 Hours**

MODULE V

Network Management and Operating System for Wireless Sensor Networks: Introduction, Network Management Requirements, Traditional Network Management Models, Network Management Design Issues. **Operating Systems for Wireless Sensor Networks:** Introduction, Operating System Design Issues, Examples of Operating Systems.
(Chapter 9: 9.1-9.5, Chapter 10: 10.1-10.3) **10 Hours**

COURSE OUTCOMES

The students shall able to:

- Existing applications of wireless sensor actuator networks
- Elements of distributed computing and network protocol design and will learn to apply these principles in the context of wireless sensor networks
- Various hardware, software platforms that exist for sensor networks
- Overview of the various network level protocols for MAC, routing, time synchronization, aggregation, consensus and distributed tracking

TEXT BOOK:

1. KAZEM SOHRABY, DANIEL MINOLI, TAIEB ZNATI, "Wireless Sensor Networks: Technology, Protocols and Applications:, WILEY , Second Edition (Indian) , 2014

REFERENCE BOOKS:

1.Ian F. Akyildiz, Mehmet Can Vuran "Wireless Sensor Networks", Wiley 2010

2. Feng Zhao & Leonidas J. Guibas, "Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

SEM IV **Year 2014-15**

Course Title: Optical Networks	**Course Code: 14SCE423**
Credits(L:T:P):4:0:0	**Core/Elective: Elective**
Type of Course: Lecture	**Total Contact Hours:50**

Course Objectives:

-To learn the basic elements of optical fiber transmission link, fiber modes configurations and structures
-To understand the different kind of losses, signal distortion in optical wave guides and other signal degradation factors
-To learn the various optical source materials, LED structures, quantum efficiency, and Laser diodes
-To learn the fiber optical receivers such as PIN APD diodes, noise performance in photo detector, receiver operation and configuration

-To learn the fiber optical network components, variety of networking aspects, FDDI, SONET/SDH and operational principles WDM

TOPICS:

MODULE I

Client Layers of the Optical Layer: SONET/SDH: Multiplexing, CAT and LCAS, Sonnet/SDH Layers, SONET Frame Structure, SONET/SDH Physical Layer , Elements of a SONET/SDH Infrastructure, Optical Transport Network: Hierarchy, Frame Structure, Multiplexing, Generic Framing Procedure Ethernet: Frame Structure, Switches, Ethernet Physical Layer, Carrier Transport IP: Routing and Forwarding, Quality of Service Multiprotocol Label Switching: Labels and Forwarding, Quality of Service, Signaling and Routing, Carrier Transport, Resilient Packet Ring: Quality of Service, Node Structure, Fairness Storage-Area Networks: Fiber Channel. **10 Hours**

MODULE II

WDM Network Elements: Optical Line Terminals, Optical Line Amplifiers, Optical Add/Drop Multiplexers: OADM Architectures, Reconfigurable OADMs Optical Cross connects: All-Optical OXC Configurations.

10 Hours

MODULE III

Control and Management: Network Management Functions: Management Framework, Information Model, Management Protocols. Optical Layer Services and Interfacing, Layers within the Optical Layer, Multivendor Interoperability, Performance and Fault Management: The Impact of Transparency, BER Measurement, Optical Trace, Alarm Management, Data Communication Network (DCN) and Signaling, Policing, Optical Layer Overhead, Client Layers. Configuration Management: Equipment Management, Connection Management, Adaptation Management. Optical Safety: Open Fiber Control Protocol

10 Hours

MODULE IV

Basic Concepts: Protection in SONET/SDH: Point-to-Point Links, Self-Healing Rings, Unidirectional Line-Switched Rings, Bidirectional Line-Switched Rings, Ring Interconnection and Dual Homing. Protection in the Client Layer: Protection in Resilient Packet Rings, Protection in Ethernet, Protection in IP, Protection in MPLS, Why Optical Layer Protection: Service Classes Based on Protection. Optical Layer Protection Schemes: 1+1 OMS Protection, 1:1 OMS Protection, OMS-DPRing, OMS-SPRing, 1:N Transponder Protection, 1+1 OCh Dedicated Protection, OCh-SPRing, OCH-Mesh Protection, GMPLS Protection, Interworking between Layers.

10 Hours

MODULE V

WDM Network Design: Cost Trade-OFFS: A Detailed Ring Network Example LTD and RWA Problems, Light path Topology Design, Routing and Wavelength Assignment, Wavelength Conversion. Dimensioning Wavelength-Routing Networks, Statistical Dimensioning Models: First-Passage Model, Blocking Model, Maximum Load Dimensioning Models: Offline Light path Requests, Online RWA in Rings.

10 Hours

Course Outcomes:

The students shall able to:

- Gain Knowledge on fundamentals of optical network.

- Explore optical network architectures ranging from optical access networks to backbone optical transport networks.
- Choose approaches and methodologies of optical network for design effective optimization;
- Apply Techniques of optical network survivability.
- Gain knowledge on Problem solving skills and critical thinking in the discipline of optical networks.

TEXT BOOK:

1. Optical Networks by Rajeev Ramaswamy, Kumar N Sivarajan, Galen H Sasaki, Elsevier Publication 3rd Edition, 2009.

References:

1. Uyless Black,Optical Networks-Third generation transport system: Pearson 2013.

SEM IV **Year 2014-15**

Course Title: Enterprise Application Programming	Course Code: **14SCE424**
Credits(L:T:P):4:0:0	Core/Elective: Elective
Type of Course: Lecture	Total Contact Hours:50

COURSE OBJECTIVES:

- To gain knowledge about metrics Web Application Development and related terminologies
- To gain knowledge about persistent framework and other ORM tools.
- To learn to build solutions using Design Patterns
- To get introduced to latest WEB frameworks

TOPICS

MODULE I

Web application and java EE 6: Exploring the HTTP Protocol, Introducing web applications, describing web containers, exploring web architecture models, exploring the MVC architecture. **Working with servlets 3.0**Exploring the features of java servlet, Exploring new features in servlet 3.0, Exploring the servlet API, explaining the servlet life cycle, creating a sample servlet, creating a servlet by using annotation, working with servletconfig and servletcontext objects, working with the Httpservlet request and HttpHttpservlet response interfaces, Exploring request delegation and request scope, implementing servlet collaboration.

10 hours

MODULE II

Handling sessions in servlet 3.0: Describing a session, introducing session tracking, Exploring the session tracking, mechanisms, using the java servlet API for session tracking, creating login application using session tracking. **Implementing event handling** Introducing events, Introducing event handling, working with the servlet events, developing the online shop web application. **Working with java server pages:** Introducing JSP technology, Exploring new features of JSP2.1, listing advantages of JSP over java servlet, Exploring the architecture of a JSP page, Describing the life cycle of a JSP page, working

with JSP basic tags and implicit objects, working with the action tags in JSP, exploring the JSP unified EL, using functions with EL.

10 hours

MODULE III

Implementing JSP tag extensions: Exploring the elements of tag extensions, Working with classic tag handlers, Exploring the tag extensions, Working with simple tag handlers. **Implementing java server pages standard tag library 1.2:** Introducing JSTL, Exploring the tag libraries JSTL, working with the core tag library. **Implementing filters:** Exploring the need of filters, exploring the working of filters, exploring filters API, configuring a filter, creating a web application using filters, using initializing parameter in filters.

10 hours

MODULE IV

Persistence Management and Design Patterns: Implementing java persistence using hibernate Introducing hibernate, exploring the architecture of hibernate, downloading hibernate, exploring HQL, understanding hibernate O/R mapping, working with hibernate, Implementing O/R mapping with hibernate. **Java EE design patterns:** Describing the java EE application architecture, Introducing a design patterns, discussing the role of design patterns, exploring types of patterns.

10 hours

MODULE V

Web Frameworks: Working with struts 2 Introducing struts 2, understanding actions in struts 2.**Working with java server faces 2.0:** Introducing JSF, Explaining the features of JSF, Exploring the JSF architecture, describing JSF elements, Exploring the JSF request processing life cycle. **Working with spring 3.0:** Introducing features of the spring framework, exploring the spring framework architecture, exploring dependency injection & inversion of control, exploring AOP with spring, managing transactions. **Securing java EE 6 applications:** Introducing security in java EE 6, exploring security mechanisms, implementing security on an application server.

10 hours

COURSE OUTCOMES:

The students shall able to:

- Implement a WEB application.
- Manage deployment configurations are
- Implement Security mechanisms

Text Book:

1. Kogent learning solution: JAVA SERVER PROGRAMMING JAVA EE6(J2EE 1.6), Dreamtech press 2014