M.Sc. (Computer Science)

Part - II / Semester 4

(CORE) CS 401: Full Time Industrial Training/ Industrial Project

Period – Minimum 4 months

1. There will be a teacher coordinator for a group of students. A teacher coordinator will take care of joining letters from students along with other necessary submission listed below.

2. A student will have to submit 2 reports during the period of ITP to the Department of the college.

3. After the completion of the ITP, a student will have to submit a synopsis along with the project completion certificate from the respective industry/research institute /educational institute.

4. A student will submit one hard copy (Student Copy) and a soft copy's (preferably 2 CDs) of the work carried out towards ITP.

5. The project will be graded by the experts (One internal examiner, one external examiner(academic expert) and one industrial expert) as follows:

O – 75 and above	**C**– 50 and above	**F-** A student will have to carry out project once again for a complete semester
A – 65 and above	**D**– 45 and above	
B – 55 and above	**E**– 40 and above	

Important Note: A student can complete ITP with a research project of a teacher / an expert funded by the University of Pune/ a funding agency.

Evaluation for internal 50 Marks will be done according to Progress Report written by Teacher Coordinator

Evaluation for external 50 Marks will be done by Industrial Expert, Academic Expert and One Internal Examiner.

(ELECTIVE) CS 402: Parallel Computing

No. of lectures: 48

Pre-requisites

- Ability to program well in C, C++ or Fortran.
- Willingness to rethink how problems should be solved.
- Algorithm & Data Structures
- Basics of Computer Architecture

Objectives

- Learning basic models of parallel machines and tools
- How to parallelize programs and how to use basic tools like MPI and POSIX threads.

Chapter 1 : Introduction to Parallel Computing [6]

Why Parallel Computing & Scope of Parallel Computing, Sieve of Eratosthenes, Control and Data Approach, PRAM model of parallel computation, Design paradigmsof Parallel Computing, examples, Bulk Synchronous Parallel (BSP) model.

Chapter 2 : Classification [12]

Flynn's Taxonomy, MPP, SMP, CC-NUMA, Clustering of Computers, Beowulf Cluster, Use of MPI in Cluster Computing. Debugging, Evaluating and tuning of Cluster Programs, Partitioning and Divide and Conquer Strategies. Cluster: dedicated high performance (HP), high availability (HA), CoPs, PoPs, CoWs; distributed, on-demand, high-throughput, collaborative, data-intensive computing, Interconnection networks.

Chapter 3 : An overview of Parallel Programming Paradigms [10]

Foster's design paradigm for Multi computing programming, Programmability Issues, Programming Models: Message passing, Message passing standards: PVM (Parallel Virtual Machine), MPI (Message Passing Interface) and its routines, Advanced Features of MPI

Chapter 4 : Overview of Programming with Shared Memory [12]

Overview of Programming with Shared Memory: OpenMP (History, Overview, Programming Model, OpenMP Constructs, Performance Issues and examples, Explicit Parallelism: Advanced Features of OpenMP)

Chapter 5 : Multi-Core programming [8]

Multi-Core programming: Introduction to Multi cores Programming Software Multi-threading using Tread Building Blocks (TBB) and Cilk++ programming, GPGPU programming with CUDA

Reference Books

1. Quinn, M. J., Parallel Computing: Theory and Practice (McGraw-Hill Inc.).
2. Bary Wilkinson and Michael Allen: Parallel Programming Techniques using Networked of workstations and Parallel Computers, Prentice Hall, 1999.
3. R. Buyya (ed.) High Performance Cluster Computing: Programming and Applications, Prentice Hall, 1999.
4. William Gropp, Rusty Lusk, Tuning MPI Applications for Peak Performance, Pittsburgh (1996).
5. W. Gropp, E. Lusk, N. Doss, A. Skjellum, A high performance portable implementation of the message passing Interface (MPI) standard, Parallel Computing 22 (6), Sep 1996.
6. Gibbons, A., W. Rytter, Efficient Parallel Algorithms (Cambridge Uni. Press).
7. Shameem A and Jason, Multicore Programming, Intel Press, 2006.
8. CUDA Programming A Developer's Guide to Parallel Computing with GPUs Shane Cook, Morgan Kaufmann

(ELECTIVE) CS 403: Embedded System

No of Lectures: 48

Pre-requisites

- Knowledge of microprocessor architecture and assembly language, microprocessor peripherals, digital design, and the C programming language is a prerequisite for this course.
- An understanding of compilers, assemblers, linkers, operating systems, analog design, diodes, transistors, and electromagnetic fields and waves will be useful

Objectives

- Understand and design embedded systems and real-time systems
- For real-time systems:
  - Identify the unique characteristics of real-time systems
  - Explain the general structure of a real-time system
  - Define the unique design problems and challenges of real-time systems
- Apply real-time systems design techniques to various software programs.
- For embedded systems, it will enable you to :
  - Understand the basics of an embedded system
  - Program an embedded system
  - Design, implement and test an embedded system.

Chapter 1 : Introduction to ES **[2]**

- What is ES
- Examples of ES
- Inside ES : processor, memory, peripherals, software

Chapter 2 : Embedded Processors , Memories & Peripherals **[6]**

- Microcontrollers 8051
- Discrete processors : 8-bit architecture, 16/32 bit CISC, RISC, DSP
- Integrated processors : ARM RISC
- Choosing a processor
- Memory systems : types (SRAM, DRAM, FLASH), organization, access time, validating the contents of memory
- Basic peripherals : parallel ports, timers, clocks

Chapter 3 : Real time system concepts **[12]**

- Foreground/ background systems
- Critical section of code
- Resourse, shared resourse
- Multitasking, task, task switch

• Kernel, scheduler, non-preemptive kernel, preemptive kernel
• Reentrancy, round-robin scheduling
• Task priority, static priority, dynamic priority, priority inversions, assigning task priorities
• Mutual exclusion, deadlock, synchronization, event flags, intertask communication
• Interrupts : latency, response, recovery, ISR processing time, NMI
(For 'C' implementation of above concepts, please refer to chapters 4,5,6,7 of the book "An Embedded Software Primer" by David E. Simon published by Pearson Educations)

Chapter 4 : Writing software for embedded systems [8]

• The compilation process : compile, link, load
• Cross compilers
• Run-time-libraries : processor dependent, I/O dependent, system calls,exit routines
• Writing a library, using alternative libraries
• Porting Kernels
• C extensions for embedded systems
• Buffering and other data structures
Linear buffers, Directional buffers, Double buffering, Buffer exchange, Linked lists, FIFO, Circular buffers, Buffer underrun and overrun, Allocating buffer memory, Buffer leakage
• Downloading

Chapter 5 : Emulation and Debugging techniques [6]

- Debugging techniques :
  - HLL simulation, low level simulation, on-board debugger, task level debugging, symbolic debug
- Emulation
- Optimization problems

Chapter 6 : Basic design using RTOS [6]

• Overview
• Principles
• Example
• Encapsulating semaphores and queues
• Hard real time scheduling considerations
• Saving memory space
• Saving power

Chapter 7 : Real time without RTOS [8]

• Choosing the SW environment
• Deriving real time performance from non-real time system
• Scheduling and data sampling
• Controlling from an external switch
• Problems

Reference Books

1. Embedded Systems Design – Steve Heath
2. Programming Embedded Systems – Michael Barr
3. Embedded Systems Building Blocks _ Jean J. Labrosse
4. An Embedded Software Primer _ David E. Simon published by Pearson Educations

(ELECTIVE) CS 404: Software Quality Assurance

No of lectures: 48

Pre-requisites

- Basic concepts of software testing

Objectives

- To enable student to learn Software Quality Assurance good practices with the help of various techniques, Strategies and tools

Chapter 1 : Software quality [4]

- Definition
- Software errors, software faults and software failures
- Software quality assurance – definition and objectives
- Software quality assurance vs. software quality control
- The objectives of SQA activities

Chapter 2 : Pre-project SQA Components [4]

- Contract Review
- Development and Quality Plan

Chapter 3 : SQA components in Project life cycle activities assessment [10]

- Verification and Validation
- Various types of Reviews
- Inspections
- Walkthrough
- Software testing
- Impact of CASE Tools

Chapter 4 : SQA Infrastructure Components [8]

- Procedures and procedure manuals
- Templates and Checklists
- Staff training
- Corrective and preventive actions
- Documentation control

Chapter 5 : Software Quality Factors [5]

- Mccall's Quality Model
- Product, Process quality metrics

Chapter 6 : Standardization **[4]**

- ISO 9001 and ISO 9000-3
- SEI-CMM,
- IEEE 1012 standard
- ISO/IEC 12207 standard

Chapter 7 : Configuration Management **[4]**

- Change control
- Release and version control
- Software configuration management audit

Chapter 8 : Quality Improvement Technique **[4]**

- Pareto Diagrams
- Cause-Effect Diagrams
- Scatter Diagrams
- Run Charts

Chapter 9 : Quality Costs **[5]**

- Quality Cost Measurement
- Utilizing Quality Costs for Decision-Making

Reference books

1. Software Quality Assurance from theory to implementation – Danial Galin
2. Software Project management - Edwin Bennatan
3. Software Engineering Roger S. Pressman,TMH,7Th Ed.
4. Software Quality Assurance : Principles and Practices Nina Godbole,
5. Project Management Body of Knowledge – PMI
6. www.softwarecertifications.org
7. Quality, 5th ed., Prentice-Hall, 2010. Donna C. S. Summers
8. Total Quality Management, Prentice Hall, 2003. Dale H. Besterfield
9. Software engineering: An Engineering approach, John Wiley. J.F.Peters, W.Pedrycz

Note: -

- **Group wise case studies are expected as a part of Internal Evaluation.**

(ELECTIVE) CS 405: Modeling and Simulation

No of lectures: 48

Pre-requisites

- The course assumes a previous knowledge of probability and statistics.
- Basic concepts of network topologies.

Objectives

- The purpose of this course is to provide students with an opportunity to develop skills in modeling and simulating a variety of problems.
- After learning thesimulation techniques, the students are expected to be able to solve real world problemswhich cannot be solved strictly by mathematical approaches.

Chapter 1 : Systems modeling [2]

Concepts of continuous and discrete formalisms. Stepped and Event-based Time in Simulations, Sources and Propagation of Error

Chapter 2 : Types of Simulations [4]

Graph or Network Transitions Based Simulations, Actor Based Simulations, Mesh Based Simulations, Hybrid Simulations, Framework for Simulation and Modeling,

Chapter 3 : Modeling and simulators [20]

Modeling formalisms and their simulators, discrete time, continuous time, discrete event, process based simulators. Hybrid systems and their simulators

Chapter 4 : Probability [8]

Basic probability, probability distributions, estimation, testing of hypotheses

Chapter 5 : Probability in modeling [8]

Selecting input probability distributions, models of arrival processes, Queues and Random Noise, Random number generators, their evaluation, generating random variates from various distributions

Chapter 6 : Analyzing models [6]

Output analysis, transient behavior, steady state behavior of stochastic systems, computing alternative systems, variance reduction techniques. Sensitivity Analysis, Verification and Validation

Text books

1. Discrete-Event System Simulation, Fourth Edition, Banks, by J., et.el. (2005), Publisher Pearson, ISBN-13: 9780131293427
2. Simulation Modeling and Analysis, Third Edition, by Law, A.M. and W.D. Kelton (2000), Publisher McGraw-Hill, ISBN-13: 978-0071165372

Reference Books

1. Continuous System Simulation, by Kofman and Cellier, Publisher Springer, ISBN-13: 9780387261027
2. Theory of modeling and Simulation, 2nd ed., B. Zeigler, H. Praehofer, T. Kim, Publisher Academic Press, 2000, ISBN-13: 978-0127784557
3. Modeling with Data: Tools and Techniques for Scientific Computing, by Ben Klemens, Publisher: Princeton University Press 2008, ISBN-13: 9780691133140

Note: -

- **Hands on can be taken with any simulating software.**