UNIVERSITY OF CALICUT

SCHEME AND SYLLABI

FOR

FIFTH SEMESTER

OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

FROM 2004 ADMISSION ONWARDS

CALICUT UNIVERSITY (P.O), THENHIPALAM

CS : COMPUTER SCIENCE & ENGINEERING

FIFTH SEMESTER

Code	Subject	Hours/Week			Sessional Marks	University Examination	
		L	T	P/D		Hrs	Marks
CS04 501	Software Engineering	3	1	-	50	3	100
CS04 502	Digital Data Communication	3	1	-	50	3	100
CS04 503	Operating Systems	3	1	-	50	3	100
CS04 504	Numerical Analysis and Optimization Techniques	3	1	-	50	3	100
CS04 505	Programming Paradigms	3	1	-	50	3	100
CS04 506	Theory of Computation	3	1	-	50	3	100
CS04 507(P)	Programming Paradigms Lab	-	-	3	50	3	100
CS04 508(P)	Hardware Lab	-	-	3	50	3	100
TOTAL		**18**	**6**	**6**	**400**	**-**	**800**

FIFTH SEMESTER

CS04 501 : SOFTWARE ENGINEERING

(common with IT04 501)

3 hours lecture and 1 hour tutorial per week

[Objective of the course is to introduce the software engineering techniques and background information to the students of computing sciences stream. For adequacy this has to be complemented by exercises appearing in texts and references. One suggestion is to consider using techniques learned here while doing mini project.**]**

Module I (11 hours)

Introduction:

Definition - History - Software life cycle - Software Engineering & other areas of computer science - Nature of a software product - Representative qualities - Quality requirements in different application areas - Idea of quality assurance.

Software Engineering principles - Illustrative case studies.

Module II (14 hours)

Design:

Relation of software engineering principles to design - Design activity & its objectives - Modularization techniques - module structure and its representation, interface and information hiding, categories, specific techniques to accommodate change, stepwise refinement, top-down and bottom-up design - Handling anomalies.

Concurrent software - methods to keep consistency - Real time software - Distributed software - issues in building modules, module integration - Object oriented design.

Architecture:

Standard - Components - Architecture for component integration -Architecture for distributed systems.

Specification:

The different contexts - Typical uses - Different styles - Verification of specification.

Operational specification notations - Definition with example for DFD, UML, Finite state machines - Descriptive specification notations - Definition with example of E-R diagrams, logic specification, algebraic specification - Building & using specifications in practice.

Module III (15 hours)

Verification:

Goals and requirements of verification - Approaches to verification.

Testing - Goals for testing - Theoretical foundations - Empirical testing principle - White box testing, black box testing - Top-down & bottom-up integration - Testing object oriented programs - Separate concerns in testing activity - Testing concurrent & real time systems.

Analysis - Informal techniques - Basic concepts of correctness proof - Using correctness proof in practice - Symbolic execution - Basic concepts - Model checking.

Verifying other software properties - Metrics for verifying qualities.

Production Process:

Software Process Model - Importance - Main activities in software production - feasibility study, specifying requirements, detailed design, testing, system testing, delivery & maintenance, other related activities.

Process models - Waterfall model, Evolutionary model, Transformational model, Spiral model - An assessment of process models - Dealing with Legacy software - Case study: A telephone switching system - Case study: Synchronize & stabilize process - Case Study: Open source approach.

Configuration management - Software standards.

Module IV (12 hours)

Management:

Functions - Project planning - Software productivity - Productivity metrics - Factors affecting productivity - Cost estimation - Predictive models - COCOMO & COCOMO II - Project control - Work breakdown structures, Gantt charts, PERT charts - Dealing with deviations - Team organization -

centralized, de-centralized, mixed - An assessment of team organizations - Risk management - Capability maturity model.

Tools & Environments:

Evolution - Dimensions for comparing tools - Representative tools - Tools for software testing - Static analyzers - GUI tools - Configuration management tools - Tracking tools - Reverse and re-engineering tools - Management tools - Tool integration - Evolution of tools.

Future - Role of the software engineer - Ethics and social responsibility.

Text book

1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli; Fundamentals of Software Engineering; 2nd edition; Pearson Education Asia

Reference books

1. Pressman R.S.; Software engineering - A practitioner's approach; 5th edition; McGraw Hill Higher education series.
2. Mall R.; Fundamentals of Software Engineering; Prentice Hall of India
3. Behferooz A. & Gydsib F.J.; Software Engineering fundamentals; Oxford University Press.
4. Jalote P.; An Integrated approach to Software Engineering; Narosa
5. Ian Sommervillie; Software Engineering, Pearson Education Asia

Sessional work assessment

Assignments	2x7.5 = 15
Tests	2x15 = 30
Regularity	= 05
Total marks	= 50

University examination pattern

Q I - 8 short type questions of 5 marks each, 2 from each module
Q II - 2 questions of 15marks each from module I with choice to answer any one
Q III - 2 questions of 15marks each from module II with choice to answer any one
Q IV - 2 questions of 15marks each from module III with choice to answer any one
Q V - 2 questions of 15marks each from module IV with choice to answer any one

CS04 502 DIGITAL DATA COMMUNICATION

(common with IT04 502)

3 hours lecture and 1 hour tutorial per week

[Objective: This course is useful for the students in understanding the fundamental theory associated with Data Communication, which is the basis for all forms of Computer networks. This syllabus focuses on essential principles of digital transmission and the reliable transfer of data between Computers located at various places.]

Module I (13 hours)

Data Communication – networks – protocols – standards, Basic concepts – Line configuration – topology- transmission modes, Categories of Networks, Internet works, OSI model – functions of the layers, Electronic noise – external and internal noises – signal to noise ratio – noise figure, Signals – analog and digital signals – time & frequency domains, Analog and digital data transmission, Transmission impairments, Channel Capacity - Nyquist and Shannon theorems.

Module II (13 hours)

Encoding – digital to digital – analog to digital (PAM, PPM, PWM & PCM) – digital to analog (ASK, FSK, PSK & QAM) – analog to analog, Spread spectrum, Transmission of digital data – asynchronous & synchronous transmission – DTE-DCE interface – E1A – 232 standard – Modems, Transmission media – guided media – Twisted pair, Co-axial cable & Fiber optics, Unguided media – Microwave - Satellite & cellular

Module III (13 hours)

Multiplexing – TDM – FDM, Telephone system – analog services (switched & leased), digital service (switched/56, DDS and DS), Error detection (VRC, LRC, CRC & checksum), Error correction- Hamming code, Data compression – Huffman coding – dynamic Huffman coding – Facsimile compression,

Module IV (13 hours)

Data link control – line discipline – flow control – error control (stop & wait ARQ, Go back N & selective repeat), Data link protocols, Asynchronous protocols (X-MODEM, Y-MODEM, Z-MODEM & Kermit), Synchronous protocols – character oriented protocol – Bit oriented protocols (HDLC, LAPs)

Text book

Behrouz Forouzan, Introduction to Data Communication and Networking, Tata McGraw- Hill Publishing Company Ltd.

Reference books

1. Fred Halsall, Data Communication, Computer Networks and Open System, Pearson Educations
2. William Stallings, Data & Computer Communications, Prentice Hall of India Pvt. Ltd.
3. Harold Kolimbiris, *Digital Communications Systems*, Pearson Education
4. Ray Horak, *Communications Systems and networks*, Dream – tech
5. William A Shay, Understanding Communication and Networks, Thomson Books/Cole
6. Michael A Miller, Data & Network Communications, Thomson Delmar Learning

Sessional work assessment

Assignments	2x7.5 = 15
Tests	2x15 = 30
Regularity	= 05
Total marks	= 50

University examination pattern

Q I - 8 short type questions of 5 marks each, 2 from each module
Q II - 2 questions of 15marks each from module I with choice to answer any one
Q III - 2 questions of 15marks each from module II with choice to answer any one
Q IV - 2 questions of 15marks each from module III with choice to answer any one
Q V - 2 questions of 15marks each from module IV with choice to answer any one

CS04 503 : OPERATING SYSTEMS
(common with IT04 503)

3 hours lecture and 1 hour tutorial per week

[Objective: This course is to impart the students the need and requirement of an interface between Man and Machine; to enable them to identify the difference between the system software and the application software and their design requirements. The syllabus includes the features of operating systems and the fundamental theory associated with process, memory and file managements components of operating systems.]

Module I (12 hours)

Review of operating system strategies - resources - processes - threads - objects - operating system organization - design factors - functions and implementation considerations - devices - characteristics - controllers - drivers - device management - approaches - buffering - device drivers - typical scenarios such as serial communications - storage devices etc

Module II (12 hours)

Process management - system view - process address space - process and resource abstraction - process hierarchy - scheduling mechanisms - various strategies - synchronization - interacting & coordinating processes - semaphores - deadlock - prevention - avoidance - detection and recovery

Module III (12 hours)

Memory management - issues - memory allocation - dynamic relocation - various management strategies - virtual memory - paging - issues and algorithms - segmentation - typical implementations of paging & segmentation systems

Module IV (16 hours)

File management - files - implementations - storage abstractions - memory mapped files - directories and their implementation - protection and security - policy and mechanism - authentication - authorization - case study of unix kernel and Microsoft windows NT (concepts only)

Text book

1. Nutt G.J., *Operating Systems - A Modern Perspective*, Addison Wesley

Reference books

1. Silberschatz & Galvin, *Operating System Concepts*, Addison Wesley
2. Crowley C., *Operating Systems- A Design Oriented Approach*, Tata McGraw Hill
3. Tanenbaum A.S., *Modern Operating Systems*, Prentice Hall, Pearson Education

Sessional work assessment

Assignments	2x7.5 = 15
Tests	2x15 = 30
Regularity	= 05
Total marks	= 50

University examination pattern

Q I - 8 short type questions of 5 marks each, 2 from each module
Q II - 2 questions of 15marks each from module I with choice to answer any one
Q III - 2 questions of 15marks each from module II with choice to answer any one
Q IV - 2 questions of 15marks each from module III with choice to answer any one
Q V - 2 questions of 15marks each from module IV with choice to answer any one

CS04 504 : NUMERICAL ANALYSIS & OPTIMIZATION TECHNIQUES

3 hours lecture and 1 hour tutorial per week

[Objective: This course is aimed to introduce numerical method, which is unique for typical computer related applications compared to any other digital methods. Also methods to optimize various applications are introduced with suitable examples which are essential in computer simulation techniques.**]**

Module I: Numerical analysis I (10 hours)

Errors in numerical calculations - sources of errors - significant digits - numerical solution of polynomial and transcendental equations - bisection method - regula-falsi method - Newton-Raphson method - fixed point method of iteration - rates of convergence of these methods - solution of system of algebraic equations - exact methods - Crout's triangularization method - iterative methods - gauss - seidel and relaxation method - polynomial interpolation - Lagrange interpolation polynomial - divided differences - Newtons` divided difference interpolation polynomial - finite differences - operators Δ,∇,e,δ- gregory - Newton forward and backward difference interpolation polynomials - central differences - stirlings interpolation formulae

Module II: Numerical analysis II (16 hours)

Numerical differentiation - differentiation formulae in the case of equally spaced points - numerical integration - trapezoidal and Simpsons` rules - compounded rules - errors of interpolation and integration formulae numerical solution of ordinary differential equations - single step methods - Taylor series method - Eulers` method - modified Eulers` method - Picards` iteration method - runge - kutta methods (2^{nd}, 3^{rd} and 4^{th} order formulae - derivations not required) - multistep methods - Milnes` predictor and corrector formulae

Module III: Optimization techniques I (16 hours)

Optimization methods - mathematical formulation of linear programming problem - simplex method - artificial variables - Charnes M method - two phase technique - duality in linear programming - dual simplex method

Module IV: Optimization techniques II (10 hours)

Transportation assignment and routing problems

Reference books

1. Sastry S.S., *Numerical Analysis*, Prentice Hall India
2. Froberg, *Introduction to Numerical Analysis*, Second Edition, Addition Wesley
3. Salvadori & Baron, *Numerical Methods in Engineering*, Prentice Hall India
4. Gerald, *Applied Numerical Analysis*, Addison Wesley
5. Grawin W.W., *Introduction to Linear Programming*, McGraw Hill
6. Gass S.I., *Introduction to Linear Programming*, Tata McGraw Hill

Sessional work assessment

Assignments	2x7.5 = 15
Tests	2x15 = 30
Regularity	= 05
Total marks	= 50

University examination pattern

Q I - 8 short type questions of 5 marks each, 2 from each module
Q II - 2 questions of 15marks each from module I with choice to answer any one
Q III - 2 questions of 15marks each from module II with choice to answer any one
Q IV - 2 questions of 15marks each from module III with choice to answer any one
Q V - 2 questions of 15marks each from module IV with choice to answer any one

CS04 505 : PROGRAMMING PARADIGMS

3 hours lecture and 1 hour tutorial per week

[Objective: This course is to introduce the different models of programming and the various constructs and their implementation to support on a bare machine. It is hoped that the students will be familiar with principles of design of programming language after going through the course. The text given is presenting the idea using abstract notation.**]**

Module I (15 hours)

Role of programming languages- high level languages- programming paradigms-language implementation on a machine- language Syntax description- notation for expressions, abstract syntax trees, lexical syntax, context free grammars, variants of grammars- Language Semantic description- introduction to synthesized attributes, attributed grammar, natural semantics, de-notational semantics

Imperative programming: Introduction- structured programming- constructs for structured control flow- syntactic concerns- handling special cases in loops- discussion based on C. Role of types: Basic types- compound types like arrays, records, union and variant records, sets- pointers and dynamic allocation- Types and error checking- discussion based on C. Introduction to procedures: parameter passing methods- scope rules- nested scopes- implementation- discussion based on C

Module II (12 hours)

Object oriented programming: Introduction- grouping of data and operations- constructs for program structuring- information hiding- program design with modules- modules and defined types- illustration based on C++ on class declaration, dynamic allocation, templates, objects. Definition of object- object oriented thinking- Inheritance- derived classes and information hiding- illustration based on C++.

Module III (12 hours)

Functional Programming: Elements of Functional programming- Types: values and operations- Functional declaration- approaches to expression evaluation- lexical scopes- type checking. Functional programming with lists- introduction to scheme- structures of lists- list manipulation- simplification of expressions- storage allocation for lists.

Module IV (13 hours)

Logic Programming: Introduction – computing with relations- introduction to PROLOG- data structures- programming techniques- control in PROLOG- cuts.

Concurrent programming: parallelism in hardware- implicit synchronization- interleaving- liveness properties- safe access to shared data- synchronized access to shared variables.

Text book

1. Sethi R., *Programming Languages: Concepts and Constructs*, Addison Wesley

Reference books

1. Tennent R.D., *Principles of Programming Languages*, Prentice Hall International.
2. Sayed. H, Roosta; Foundation of programming languages Design and Implementations; Vikas Publising House, New Delhi.
3. Pratt T.W., and Zelkowitz M.V., *Programming Languages: Design and Implementation*, Prentice Hall International.
4. Appleby. D and J.J. VandeKopple J.J; *Programming Languages: Paradigm and Practice*, Tata McGraw-Hill
5. Scott M.L; Programming Language Pragmatics; Harcourt Asia(Morgan Kaufman).
6. Clocksin W F, Mellish C S; Programming in PROLOG

Sessional work assessment

Assignments	2x7.5 = 15
Tests	2x15 = 30
Regularity	= 05
Total marks	= 50

University examination pattern

Q I - 8 short type questions of 5 marks each, 2 from each module
Q II - 2 questions of 15marks each from module I with choice to answer any one
Q III - 2 questions of 15marks each from module II with choice to answer any one
Q IV - 2 questions of 15marks each from module III with choice to answer any one
Q V - 2 questions of 15marks each from module IV with choice to answer any one

CS04-506 : THEORY OF COMPUTATION

3 hours lecture and 1 hour tutorial per week

[Objective: This is a fundamental course on computational models and computability. It begins with the introductory concepts of languages and their classification, and proceeds through their recognizers and automata. The course concludes with a treatment on the ways and means of classifying algorithm into the various computability classes and proofs of some standard algorithms. **]**

Module I (13 hours)

Introduction to formal proof – Inductive proofs – Concepts of automata theory – Deterministic finite automata – Nondeterministic finite Automata – equivalence of deterministic and nondeterministic finite automata – Nondeterministic Finite automata with a transitions – Regular expressions – Finite automata and regular expressions – Algebraic laws for Regular expressions – Pumping lemma for regular languages – closure properties of regular languages – Decision properties of regular languages – Equivalence and minimization of automata.

Module II (13 hours)

Context free Grammars – Derivations – sentential forms – The language of grammar – Parse trees – Ambiguity in grammar and languages – Inherently ambiguous languages – Pushdown automata – Formal definition – Graphical notation – The language of a PDA – Acceptance by PDA – Empty stack – Final state – PDAs to grammars – Deterministic PDAs and CFLs – Non deterministic PDAs – Chomsky Normal Form – Greibach Normal Form – Pumping lemma for CFLs – Closure properties of CFLs – Decision properties of CFLs – CYK algorithm

Module III (14 hours)

Turing Machines – Notation – Instantaneous Description – Transition Diagram – The language of a Turing Machine – Halting of TMs – Programming techniques for Turing Machines – Extension to basic TMs – Nondeterministic TMs – Restricted TMs – Recursive and Recursively Enumerable Languages – Halting problem of TMs – Undecidable problem about TMs – Rice's Theorem – Post's Correspondence problem – Undecidablity of PCP – Undecidable problems on Languages

Module IV (12 hours)

Intractable problems – The classes P and NP – Polynomial time reducibility – NP-Complete problems – The Satisfiability problem – NP-Completeness of the satisfiability problem – NP-Completeness of CSAT – NP-Completeness of 3SAT – Node cover problem – Directed Hamiltonian circuit problem – The class of languages Co-NP – Problems solvable in polynomial space.

Text books

1. Hopcroft J.E, Motwani R & Ullman J. D., Introduction to Automata Theory, Languages and Computation, Pearson Education.

Reference books

1 Hopcroft J. E. & Ullman J. D., Introduction to AutomataTheory, Languages and Computation, Narosa
2. Linz: P., An Introduction to Formal Languages & Automata, Narosa
3. Martin J. C., Introduction to Languages & the Theory of Computation, Tata McGraw Hill

Sessional work assessment

Assignments	2x7.5 = 15
Tests	2x15 = 30
Regularity	= 05
Total marks	= 50

University examination pattern

Q I - 8 short type questions of 5 marks each, 2 from each module
Q II - 2 questions of 15marks each from module I with choice to answer any one
Q III - 2 questions of 15marks each from module II with choice to answer any one
Q IV - 2 questions of 15marks each from module III with choice to answer any one
Q V - 2 questions of 15marks each from module IV with choice to answer any one

CS04 507(P) : PROGRAMMING PARADIGM LAB

3 hours practical per week

[Objective: This Laboratory is intended to impart the working experience on paradigms of programming. This additionally serves as introduction to various programming languages representative of each paradigm. The thrust is in teaching the paradigms not the platforms. However, adequate knowledge about platform is a need for successful experimentation**]**

Lab.1: (object-oriented programming in - Java /C+ +) – programming to bring out the concept of classes and objects- for example the abstract data type binary tree.

Lab 2: (object-oriented programming) – programming to demonstrate inheritance and class hierarchy – for example define a base class "shape" and derived classes for rectangle, square, ellipse, circle with proper class hierarchy.

Lab.3: (object oriented programming) programming to demonstrate polymorphism, virtual functions- for example define base class for vectors and use inheritance to define complex and real vector with standard operations.

Lab.4:(functional programming - in Lisp) – programming to demonstrate functional specification for a solution – for example implementation of quick sort.

Lab.5:(functional programming) – programming to demonstrate implementation of conventional data structures- for example implementation of binary search tree with insertion, deletion and search operations.

Lab.6: (functional programming) - programming to demonstrate the use of available data structures in functional programming languages-for example implementation of set with membership, union and intersection operations

Lab.7: (logic programming - in prolog) – programming to demonstrate ready implementation of propositional logic statements- for example to find the gcd of two given integers

Lab.8: Lab.9: (logic programming) - programming to demonstrate language specific features- for example implementation of a logic program to check whether a given NFA accepts the given string

Lab.9: (concurrent programming- in Java) - demonstration of concurrency support- for example programming to find the least common ancestor of two given nodes in a binary tree.

Lab.10: concurrent programming- in Java) - demonstration of synchronized concurrency –

for example programming for the readers and writers problem

Reference books

1. Sethi R., *Programming Languages: Concepts and Constructs*, Addison Wesley
2. Appleby D. & Vandekopple J.J., *Programming Languages*: *Paradigm and Practice*, Tata McGraw Hill
3. Luger & Stubblefield, *Artificial Intelligence*, Addison Wesley
4. Samuel A. Rebelsky, *Experiments in Java*, Pearson Education

Sessional work assessment

Lab practicals & record	= 25
Tests	2x10 = 20
Regularity	= 05
Total marks	= 50

CS04 508(P) : HARDWARE LAB

3 hours practical per week

[Objective: This course is to teach the relevance and characteristics of hardware and operating system components of a digital computer system through various laboratory experiments. It also gives the students the ability to interface devices to computer systems through various interfacing techniques.**]**

Lab 1 : Identification of components/cards and PC assembling from components

Lab 2 : Assembly language program for implementing arithmetic operations

Lab 3,4: Implementation of a file manager using DOS/BIOS interrupts

Lab 5 : TSR (Terminate and Stay Resident) Programming

Lab 6 : ADC interface

Lab 7 : Stepper Motor interface using DAC

Lab 8,9: Parallel Interface: Printer and HEX keyboard.

Lab 10 : Serial Interface: PC to PC serial interface using MODEM.

Reference books

1. Messmer H.P., *The Indispensable PC Hardware Book*, Addison Wesley
2. Hall D.V., *Microprocessors and Interfacing*, Tata McGraw Hill
3. Norton P., *Dos Internals*

Sessional work assessment

Lab practicals and record	= 25	
Test		= 20
Regularity		= 05
Total marks		= 50