

**UNIVERSITY OF CALICUT**

**SCHEME AND SYLLABUS**

**FOR**

**FIFTH SEMESTER**

**OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

*Basil. P.T*

*AIO-H Engineering College Calicut*

*basilonline007@gmail.com*

# ITO4 501: SOFTWARE ENGINEERING

## Objective:

*Objective of the course is to quickly introduce the software engineering techniques and background information in a single paper useful for the computing sciences stream. Note that software engineering has grown to the level of becoming a course by itself; since we are covering material in a single paper because of lack of slots, the thrust of this course is on teaching existing practice. For adequacy this has to be complemented by exercises appearing in texts and references. Books have been carefully chosen to get examples from diverse computing application for practice along with theory. Those interested can find additional training materials downloadable from sites associated with each book. 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, decentralized, 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 books**

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

#### **Reference books**

*Pressman R.S.; Software engineering - A practitioner's approach; 5th edition; McGraw Hill Higher education series.*

*Mall R.; Fundamentals of Software Engineering; Prentice Hall of India*

*Behferooz A. & Gydsib F.J.; Software Engineering fundamentals; Oxford University Press.*

*Jalote P.; An Integrated approach to Software Engineering; Narosa*

*Ian Sommerville; Software Engineering, Pearson Education Asia*  
*Internal work assessment*

# IT04 502 : DIGITAL DATA COMMUNICATION

## 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 - standards - ISO reference model - internal architecture - protocol implementation issues - transmission media - attenuation and distortion - limited bandwidth - signal types - propagation delay - public carrier circuits - modulation - multiplexing - physical layer interfacing standards

## **Module - II: (14 hours)**

Data transmission basics - transmission modes - asynchronous and synchronous transmission - bit - character and frame synchronization - coding - error detection methods - parity - block sum check - cyclic redundancy check - data compression - Huffman coding - dynamic Huffman coding - facsimile compression - transmission control circuits - communication control devices

## **Module - III: (12 hours)**

Protocol basics - error control - stop-and-wait & sliding window protocol - link utilization - selective repeat and go-back-N - link management

## **Module - IV: (13 hours)**

Data link control protocols - character-oriented protocols - half-duplex protocols - duplex protocols - bit-oriented protocols - high level data link control (HDLC) - LAPB - LAPD - logical link control - protocol operation

## **Text book**

*Halsall F., Data Communication, Computer Networks and Open Systems, Addison Wesley*

## **Reference books**

*Forouzan B., Introduction to Data Communication and Networking, Tata McGraw Hill*

*William Stallings, Data and Computer Communications, PHI*

*Prakash C Gupta, Data Communications, PHI*

# ITO4 503: OPERATING SYSTEMS

## 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**

*Nutt G.J., Operating Systems - A Modern Perspective, Pearson Education Asia*

## **Reference books**

*Silberschatz & Galvin, Operating System Concepts, Pearson Education Asia*

*Crowley C. Operating Systems-A Design Oriented Approach, Tata McGraw Hill*

*Tanenbaum A.S., Modern Operating Systems, Prentice Hall of India / Pearson Education*

# ITO4 504 : INFORMATION THEORY & CODING

## **Objective:**

The subject deals with the fundamentals of information quality, error control in communication process and various systems of coding information for reliable communications. Built on a sound mathematical basis, the methods developed in this field of study are essential in a study of communication systems, information Technology and computing. A background in algebraic structures would prove helpful while learning this subject.

## ***Module - I: (14 hours)***

Information theory - information and entropy - properties of entropy of a binary memory less source - extension of a discrete memory less source - source coding theorem - Shannon-Fano coding - Huffman coding - Lempel Ziv coding - discrete memory less source - binary symmetric channel - mutual information - properties - channel capacity - channel coding theorem - information capacity theorem

## ***Module - II: (14 hours)***

Coding - linear block codes - generator matrices - parity check matrices - encoder - syndrome and error detection - minimum distance - error correction and error detection capabilities - cyclic codes - coding and decoding

## ***Module - III: (14 hours)***

Introduction to algebra - groups - fields - binary field arithmetic - construction of galois field - basic properties - computations - vector spaces - matrices - BCH codes - description - decoding - reed solomon codes

## ***Module - IV: (10 hours)***

Coding - convolutional codes - encoder - generator matrix - transform domain representation - state diagram - distance properties - maximum likelihood decoding - Viterbi decoding - sequential decoding - interleaved convolutional codes

## **Text books**

*Simon Haykin, Communication Systems, John Wiley*

*Shu Lin & Costello D.J., Error Control Coding - Fundamentals and Applications, Prentice Hall Inc. Englewood Cliffs*

## **Reference books**

*Das J., Malik S.K. & Chatterje P.K., Principles of Digital Communication, New Age International Limited*

*Sam Shanmugham, Digital and Analog Communications, John Wiley*

*Simon Haykin, Digital Communications, John*

*Taub & Shilling, Principles of Communication Systems, Tata McGraw Hill.*

# ITO4 505 : GRAPH THEORY & COMBINATORICS

## Objective

*This course comprehends the graphs as a modeling and analysis tool in computer science & Engineering. It introduces the structures such as graphs & trees and techniques of counting and combinations, which are needed in number theory based computing and network security studies in Computer Science.*

## **Module - I: (13 hours)**

Introduction to graphs - definitions - subgraphs - paths and cycles - matrix representation of graphs - euler tours - chinese postman problem - planar graphs - Euler's formula - platonic bodies - applications of Kuratowski's theorem - hamiltonian graphs - graph colouring and chromatic polynomials - map colouring

## **Module - II: (14 hours)**

Trees - definitions and properties - rooted trees - trees and sorting - weighted trees and prefix codes - biconnected components and articulation points - Kruskal's and Prim's algorithms for minimal spanning trees - Dijkstra's shortest path algorithm - bellman-ford algorithm - all-pairs shortest paths - Floyd-Warshall algorithm - the max-flow min-cut theorem - maximum bipartite matching

## **Module - III: (11 hours)**

Fundamental principles of counting - permutations and combinations - binomial theorem - combinations with repetition - combinatorial numbers - principle of inclusion and exclusion - derangements - arrangements with forbidden positions

## **Module - IV: (14 hours)**

Generating functions - partitions of integers - the exponential generating function - the summation operator - recurrence relations - first order and second order – non homogeneous recurrence relations - method of generating functions

## **Text book**

*Grimaldi R.P., Discrete and Combinatorial Mathematics: An Applied Introduction, Addison Wesley*

## **Reference books**

*Clark J. & Holton D.A., A First Look at Graph Theory, Allied Publishers (World Scientific)*

*Corman T.H., Leiserson C.E. & Rivest R.L., Introduction to Algorithms, Prentice Hall India*

*Mott J.L., Kandel A. & Baker T.P., Discrete Mathematics for Computer Scientists and Mathematicians, Prentice Hall of India*

*Liu C.L., Elements of Discrete Mathematics, McGraw Hill*

*Rosen K.H., Discrete Mathematics And Its Applications, McGraw Hill*

# IT04 506 : DATA MODELLING & DESIGN

## *Module - I: (10 hours)*

Overview of object oriented systems - objects - attributes - encapsulation - class hierarchy - polymorphism - inheritance - messages - history of object orientation

## *Module - II: (14 hours)*

UML - classes - attributes - and operations - class diagrams - generalizations and association constructs - composition and aggregation - collaboration diagrams - sequence diagrams - asynchronous messages and concurrent execution - state diagrams - nested states - concurrent states and synchronization - transient states - architecture and interface diagrams packages - deployment diagrams for hardware artifacts and software constructs - window-layout and window- navigation diagrams

## *Module - III: (14 hours)*

Encapsulation structure - connascence - domains of object classes - encumbrance - class cohesion - state spaces and behaviour of classes and subclasses - class invariants - preconditions and post conditions - class versus type - principle of type conformance - principle of closed behaviour - case studies

## *Module - IV: (14 hours)*

Abuses of inheritance - danger of polymorphism - mix-in classes - rings of operations - class cohesion and support of states and behaviour - components and objects - design of a component - lightweight and heavy weight components - advantages and disadvantages of using components - case studies

## **Text book**

*Page Jones M., Fundamentals of Object- Oriented Design in UML, Addison Wesley*

## **Reference books**

*Booch G., Rumbaugh J. & Jacobsons I., The Unified Modeling Language User Guide, Addison Wesley*

*Bahrami A., Object Oriented System Development, McGraw Hill*

*Rumbaugh J., Jacobson I. & Booch G., The Unified Modeling Language Reference Manual, Addison Wesley*

*Larman C., Applying UML & Patterns: An Introduction to Object-Oriented Analysis & Design, Addison Wesley*

*Pooley R. & Stevens P., Using UML: Software Engineering With Objects & Components, Addison Wesley*



# IT04 507(P) : PROGRAMMING PARADIGMS LAB

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

**Lab - 2** : (object oriented programming) Programs 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) - Programs 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 / scheme / Haskell) - Program to demonstrate functional specification for a solution - for example implementation of quick sort

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

**Lab -6** : (functional programming) - Program to demonstrate 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 / VisiCalc) - Program to demonstrate ready implementation of propositional logic statements - for example implementation of logic program to find the gcd of two given integers

**Lab - 8** : (logic programming) - Program to demonstrate language specific features - for example implement a logic program to check whether a given NFA accepts the given string

**Lab - 9** : (concurrent programming - Java / Ada) Demonstration of concurrency support - for example program to find the least common ancestor of two given nodes in a binary tree may be implemented.

**Lab - 10** : (concurrent programming) - Demonstration of synchronized concurrency - for example program for the readers and writers problem may be implemented.

## **Reference books**

*Sethi R., Programming Languages: Concepts And Constructs, Pearson Education Asia*  
*Appleby D. & Vandekopple J.J., Programming Languages: Paradigm And Practice, Tata McGrawHill*

*Luger & Stubblefield, Artificial Intelligence, Pearson Education Asia*  
*Samuel A. Rebelsky, Experiments in Java, Pearson Education Asia*

# IT04 508(P) : HARDWARE LAB

## Objective:

*This course is to understand 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

Messmer H.P., *The Indispensable PC Hardware Book*, Pearson Education Asia

Hall D.V., *Microprocessors and Interfacing*, Tata McGraw Hill

Norton P., *Dos Internals*

---



*A NEW WINDOW TO THE  
LANDSCAPE*