UNIVERSITY OF CALICUT

FACULTY OF ENGINEERING

Curriculum, Scheme of Examinations & Syllabi for the

MASTER OF COMPUTER APPLICATION

Degree Programme
with effect from Academic Year 2001-2002

MASTER OF COMPUTER APPLICATION

Course Structure and Scheme of Evaluation
(With effect from Academic Year 2001-2002)

III SEMESTER:

Course No	Course Title	Hours Per Week			University Exam		Sessional Marks	Total Marks
		L	**T**	**P**	Duration (Hrs)	Marks		
MCA2K 301	Numerical Analysis & Optimization Techniques	3	1	0	3	100	50	150
MCA2K 302	Operating Systems	3	1	0	3	100	50	150
MCA2K 303	Database Management Systems	3	1	0	3	100	50	150
MCA2K 304	Principals of Compilers	3	1	0	3	100	50	150
MCA2K 305	Software Engineering	3	1	0	3	100	50	150
MCA2K 306P	Software Application Lab	0	0	3	--	--	50	50
Total		**15**	**5**	**3**		**500**	**300**	**800**

MCA2K 301 : NUMERICAL ANALYSIS & OPTIMIZATION TECHNIQUES

3 hours lecture and 1 hour tutorial per week

Module I: (8 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

Module II: (10 hours)
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 III: (10 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 stepmethods - 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 IV: (14 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, transportation and Assignment problems

Module V: (10 hours)
Queuing Models: Basic Structures of queuing models, Roles of the Poisson and Exponential distributions,(M/M/I),(GD/∞ /∞),(M/M/I): (GD/N/∞),(M/M/C): (GD/∞/∞),(M/M/C)(GD/N/∞)models,Self Service Model, Machine Service Model.

Reference books

1. Sastry S.S., *Numerical Analysis*, Prentice Hall India
2. Froberg, *Introduction to Numerical Analysis*, 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	2x10 = 20
Tests	2x15 = 30
Total marks	=50

University examination pattern

Seven questions covering all the five modules. Each carries 20 marks. There should be a minimum of one question from each module. There should not be more than two questions from any module. The student has to answer any five full questions for scoring full marks.

MCA 302: OPERATING SYSTEMS

3 hours lecture and 1 hour tutorial per week

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 (8 hours)
File management - files - implementations - storage abstractions - memory mapped files - directories and their implementation

Module IV (8 hours)
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	2x10 = 20
Tests	2x15 = 30
Total marks	=50

University examination pattern
Seven questions covering all the five modules. Each carries 20 marks. There should be a minimum of one question from each module. There should not be more than two questions from any module. The student has to answer any five full questions for scoring full marks.

MCA2K 303 : DATABASE MANAGEMENT SYSTEMS

3 hours lecture and 1 hour tutorial per week

Module I (12 hours)

Introduction: characteristics of database approach - advantages of using DBMS - database concept and architecture - data models - schemes - instances - data independence - database languages and interfaces - database modeling using entity - relationship (ER) - entity sets attributes and keys - relationships - type role and structural constraints - weak entity types - enhanced entity-relationship (EER) and object modeling - sub classes - super classes and inheritance - specialization and generalization - modeling of union types

Module II (10 hours)

File organization and storage: secondary storage devices - RAID technology - operations in files - heap files and sorted files - hashing techniques - types of single level ordered index, multi-level indexes - B - trees and B^+ trees - indexes on multiple keys - other types of indexes

Module III (14 hours)

Database design: functional dependencies - normal forms - general definition of second and third normal forms - Boyce-Codd normal form - multi valued dependencies and fourth normal form - join dependencies and fifth normal form - inclusion dependencies - practical database design tuning - database design process relational model concepts - relational algebra operations - queries in SQL - insert - delete and update statements in SQL views in SQL

Module IV (8 hours)

Transaction processing: desirable properties of transactions, schedules and recoverability - serializability of schedules concurrency control - locking techniques - time stamp ordering multi version concurrency control - granularity of data items

Module V (8 hours)

Database recovery techniques based on deferred up data and immediate updating - shadow pages - ARIES recovery algorithm - database security and authorization - security issue access control based on granting/revoking of privileges introduction to statistical database security

Text book

Elmasri & Navathe, "Fundamentals of Database Systems", Addison Wesley

Reference books

Ramakrishnan R. & Gehrke J., "Database Management Systems", McGraw Hill

Silberschatz A., Korth H.F. & Sudarshan S., "Database System Concepts", Tata McGraw Hill

Ullman J.D., “Principles of Database Systems”, Galgotia Publications

Date C.J., “ An Introduction to Database Systems “, Addison Wesley

Sessional work assessment

Assignments	2x10 = 20
Tests	2x15 = 30
Total marks	= 50

University examination pattern

Seven questions covering all the five modules. Each carries 20 marks. There should be a minimum of one question from each module. There should not be more than two questions from any module. The student has to answer any five full questions for scoring full marks.

MCA2K 304 : PRINCIPLES OF COMPILERS

3 hours lecture and 1 hour tutorial per week

Module I (10 hours)
Introduction - analysis of the source program - phases of a compiler - compiler construction tools - lexical analysis - role of the lexical analyser - specification of tokens - recognition of tokens - lexical analyzer generators

Module II (15 hours)
Syntax analysis: role of the parser - context-free grammars - top-down parsing - bottom-up parsing - operator precedence parsing - LR parsers (SLR, canonical LR, LALR) - parser generators

Module III (13 hours)
Syntax-directed translation - syntax-directed definitions - S-atributed definitions - L-attributed definitions - bottom-up and top-down translation - type checking - type systems - specification of a type checker - run-time environments - source language issues - storage organization - storage allocation strategies - access to non-local names - parameter passing - symbol tables

Module IV (6 hours)
Intermediate code generation - intermediate languages - declarations - assignment statements - Boolean expressions - procedure calls

Module V (8 hours)
Introduction to code optimization - sources of optimization - introduction to data-flow analysis - introduction to code generation - issues in the design of a code generator - the target machine - a simple code generator

Text book
Aho A.V., Sethi R. & Ullman J.D. *Compilers Principles, Techniques and Tools,* Addison Wesley

Reference books
1. Aho A.V. & Ullman J.D. *Principles of compiler Design*, Narosa
2. Muchnick S.S., *Advanced Compiler Design Implementation*, Harcourt Asia (Morgan Kaufman)
3. Holub A.I., *Compiler Design in C*, Prentice Hall India
4. Appel A.W., *Modern Compiler Implementation in C*, Cambridge University Press

Sessional work assessment

Assignments	2x10 = 20
Tests	2x15 = 30
Total marks	= 50

University examination pattern
Seven questions covering all the five modules. Each carries 20 marks. There should be a minimum of one question from each module. There should not be more than two questions from any module. The student has to answer any five full questions for scoring full marks.

MCA2K 305 : SOFTWARE ENGINEERING

3 hours lecture and 1 hour tutorial per week

Module I (10 hours)

Introduction - FAQs about software engineering - professional and ethical responsibility - system modeling - system engineering process - the software process - life cycle models - iteration - specification - design and implementation - validation - evolution - automated process support - software requirements - functional and non-functional requirements - user requirements - system requirements - SRS - requirements engineering processes - feasibility studies - elicitation and analysis - validation - management - system models - context models - behavior models - data models - object models - CASE workbenches

Module II (10 hours)

Software prototyping - prototyping in the software process - rapid prototyping techniques - formal specification - formal specification in the software process - interface specification - behavior specification - architectural design - system structuring - control models - modular decomposition - domain-specific architectures - distributed systems architecture

Module III (8 hours)

Object-oriented design - objects and classes - an object oriented design process case study - design evolution - real-time software design - system design - real time executives - design with reuse - component-based development - application families - design patterns - user interface design - design principles - user interaction - information presentation - user support - interface evaluation

Module IV (11 hours)

Dependability - critical systems - availability and reliability - safety - security - critical systems specifications - critical system development - verification and validation - planning - software inspection - automated static analysis - clean room software development - software testing - defect testing - integration testing - object-oriented testing - testing workbenches - critical system validation - software evolution - legacy systems - software change - software maintenance - architectural evolution - software re-engineering - data re-engineering

Module V (13 hours)

Software project management - project planning - scheduling - risk management - managing people - group working - choosing and keeping people - the people capability maturity model - software cost estimation - productivity estimation techniques - algorithmic cost modeling, project duration and staffing quality management - quality assurance and standards - quality planning - quality control - software measurement and metrics - process improvement - process and product quality - process analysis and modeling - process measurement - process CMM - configuration management - planning - change management - version and release management - system building - CASE tools for configuration management

Text book

Ian Sommerville, *Software Engineering*, Pearson Education Asia

Reference books

1. Pressman R.S., *Software Engineering*, McGraw Hill
2. Mall R., *Fundamentals of Software Engineering*, Prentice Hall of India
3. Behferooz A. & Hudson F.J., *Software Engineering Fundamentals*, Oxford University Press
4. Jalote P., *An Integrated Approach to Software Engineering*, Narosa

Sessional work assessment

Assignments	2x10 = 20
Tests	2x15 = 30
Total marks	= 50

University examination pattern

Seven questions covering all the five modules. Each carries 20 marks. There should be a minimum of one question from each module. There should not be more than two questions from any module. The student has to answer any five full questions for scoring full marks.

MCA 306(P) : SOFTWARE APPLICATION LAB

3 hours practical per week

Database management systems

Conversion of a given relational scheme to 3NF and BCNF
Implementation of B tree and B+ tree
Implementation of a database stored in an RDBMS accessible through a web browser
Program to convert SQL subset into relational algebra (tools like YACC may be used)
Implementation of optimistic concurrency control algorithm

Software Engineering

Exercise on software modeling using CASE tools
Exercise on Software Design using CASE tools
Two exercises on Software testing
Exercises on reverse engineering of C++/Java codes
Exercises on automatic skeleton code generation for software design

Reference books

1. Ian Sommerville, Software Engineering,6/e ,Pearson education
2. Elmasri & Navathe, "*Fundamentals of Database Systems*", Addison Wesley
3. Ramakrishnan R. & Gehrke J., "*Database Management Systems*", McGraw Hill

Sessional work assessment

Laboratory practical and record	= 30	
Test/s		= 20
Total marks		= 50