



CHRIST
UNIVERSITY

BANGALORE, INDIA

Declared as Deemed to be University under Section 3 of UGC Act 1956

Department of Computer Science

Proposed Syllabus for Master of Computer Applications 2013

Course Structure

<i>Subject</i>	Semesters- No. of hours (credits)						<i>Hours</i>	<i>Credits</i>	<i>Marks</i>
	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>	<i>VI</i>			
Theory Papers	24(18)	24(18)	20(15)	20(15)	20(15)		108	81	2700
Practical Papers	08(04)	08(04)	12(06)	12(06)	12(06)		52	26	1300
Industry Project						30(06)	30	06	300
Seminar			02(01)	02(01)			04	02	100
Holistic Education	01(01)	01(01)					02	02	
Total	33(23)	33(23)	34(22)	34(22)	32(21)	30(06)	196	117	
Marks	800	800	850	850	800	300			4400
Innovative Project		(02)						02	
Total Hours and Credits							196	119	

I Semester

Paper Code	Paper	Hours per Week	Credits	Marks
MCA131	Programming using C	04	03	100
MCA132	Web Technologies	04	03	100
MCA133	Digital Logic	04	03	100
MCA134	Software Engineering	04	03	100
MCA135	Probability and Statistics	04	03	100
MCA136	Human Resource Management	04	03	100
MCA151	C Programming Lab	04	02	100
MCA152	Web Technologies Lab	04	02	100
HOL01	Holistic Education	01	01	-
Total		33	23	800

II semester

Paper Code	Paper	Hours per Week	Credits	Marks
MCA231	Microprocessors and Interfacing Techniques	04	03	100
MCA232	Object Oriented Programming using C++	04	03	100
MCA233	Operating Systems	04	03	100
MCA234	Relational Database Management System	04	03	100
MCA235	Discrete Mathematical Structures	04	03	100
MCA236	Accounting & Financial Management	04	03	100
MCA251	Assembly Language Programming Lab	04	02	100
MCA252	C++ Lab	04	02	100
HOL02	Holistic Education	01	01	-
Total		33	23	800

III semester

Paper Code	Paper	Hours per Week	Credits	Marks
MCA331	Data Structures	04	03	100
MCA332	Unix Operating System	04	03	100
MCA333	Computer Architecture	04	03	100
MCA334	OOAD with UML	04	03	100
MCA335	Computer Networks	04	03	100
MCA351	Data Structures Lab	04	02	100
MCA352	Unix Lab	04	02	100
MCA353	RDBMS Lab	04	02	100
MCA371	Seminar	02	01	50
Total		34	22	850

IV semester

Paper Code	Paper	Hours per Week	Credits	Marks
MCA431	Programming in Java	04	03	100
MCA432	Design and Analysis of Algorithms	04	03	100
MCA433	Mobile Applications	04	03	100
Elective I (Choose any one)				
MCA441A	Digital Image Processing	04	03	100
MCA441B	Advanced Database Management system			
MCA441C	Multimedia System and Applications			
MCA441D	Software Quality and Testing			
MCA441E	Microcontroller and Applications			
Elective II (Choose any one)				
MCA442A	Web Engineering	04	03	100
MCA442B	Network Security			
MCA442C	Data Warehousing			
MCA442D	Linux Administration			
MCA442E	Advanced Microprocessors			
MCA451	Java Programming Lab	04	02	100
MCA452	Computer Architecture Project Lab	04	02	100
MCA453A	Digital Image Processing Lab	04	02	100
MCA453B	ADBMS Lab			
MCA453C	Multimedia Lab			
MCA453D	Software Quality and Testing Lab			
MCA453E	Microcontroller Lab			
MCA471	Seminar	02	01	50
Total		34	22	850

V semester

Paper Code	Paper	Hours per Week	Credits	Marks
MCA531	Computer Graphics with Open GL	04	03	100
MCA532	Artificial Intelligence	04	03	100
MCA533	System Software	04	03	100
Elective III (Choose any one)				
MCA541A	Software Architecture	04	03	100
MCA541B	Wireless and Mobile Networks			
MCA541C	Parallel Computing with Open CL			
MCA541D	Machine Learning			
MCA541E	Embedded Programming and RTOS			
Elective IV (Choose any one)				
MCA542A	Information Retrieval and Web Mining	04	03	100
MCA542B	Database Administration			
MCA542C	Cloud Computing			
MCA542D	Bioinformatics			
MCA542E	Principles of User Interface Design			
MCA551	Computer Graphics Lab	04	02	100
MCA552	Computer Networks Project Lab	04	02	100
MCA553	Specialization Project Lab	04	02	100
Total		32	21	800

VI semester

Paper Code	Paper	Hours per Week	Credits	Marks
MCA651	Industry Project	30	6	300

- Total Marks : **4400**
- Total Credits : **119** (including 2 credits of Innovative project)

QUESTION PAPER PATTERN FOR ALL COMPUTER SCIENCE

THEORY PAPERS

- Question paper has to be set for the total marks of 100.
- Examination duration is 3 hours.
- The syllabus is divided into five major units.
- From each major unit, 2 full questions with internal choice have to be set.
- Each full question carries 20 marks.
- There can be maximum of 3 sub divisions in each full question.

MCA131 - Programming in C

Total teaching Hours/Semester: 60

No of Lecture Hours/Week: 04

Course objective

Understand the concept of a C program like variables, control structures, arrays, functions, pointers, macro processor, files. Understand the concepts of assembly level support by C, Graphics programming and Mouse programming in windows environment

Learning outcome

Upon successful completion of the courses in this discipline, the student will have acquired the following knowledge and skills:

- Understand the use of structured program development in C as applied to both large software systems and to small programming projects.
- Understand the use of arrays, functions, pointers, macro processors, structures, unions, files
- Understand the use and structure Graphics and mouse programming in C
- Understand the assembly language support of C

Unit I.

(12)

Introduction to C Language

Applications of C – Language Features – Identifiers - Data Types – Typecasting- variables – constants. Operators - I/O Statements : Formatted- Unformatted. Control Structures.

Unit II.

(12)

Functions

User-defined

functions – Standard library functions (Header files) - Function prototypes – Call-by-Value – Command Line Arguments, Concept of variable number of arguments

Storage Types

Introduction to Storage Types – Static, Auto, Register, Extern

Arrays

Introduction to Arrays – Limitations of Arrays – Types – Strings- I/O functions – String functions – Memory formatting (scanf & printf)- Passing arrays to functions

Unit III.

(12)

Pointers

Definition – Pointer variables – Accessing variables through pointers – pointer declaration and definition – Initialization - Pointers and Functions – Pointer to pointers – Pointer Applications - - Introduction to Dynamic memory allocation functions (malloc, calloc, free, realloc) - Array of pointers

Derived Types

Type definition (typedef) – Enumerated type – Structures – Accessing – Complex structure – Array of structures – structures & functions – Union - Use of pointers to Structures and Unions

Macro Processor

Specialty of macro processing – Declaration, Conditional, Include directives

Unit IV.**(11)****External storage**

Text files: Concept of Files – Files and Streams – Standard library I/O functions – Character I/O functions.

Binary files: Operations – Standard library functions – Converting file type – Examples

Unit V.**(13)****Operations on Bits**

Introduction to Bit-Fields – Operators – showbits() function -Assembly language applications – looping and comparison – shifting bits

C under windows

Features – Graphics– Initialization Lines – Images – Patterns – Regular and non regular shapes – palettes – colors – text – justification of text – animation

Text Books:

- [1] Forouzon A Behrouz , Gilberg F Richard ,*A Structured Programming Approach using C- 3rd Illustrated Edition*,2009
- [2] Kanetkar Yeshwant, *Let Us C*, BPB publications, 10th Edition,2010.

Reference Books:

- [1] Deitel & Deitel, *C – How to Program*, Pearson Education Asia, 6th Edition, 2010
- Gottfried Byron, *Programming with C*, Tata McGraw Hill
- [2] Kanetkar Yeshwant, *Understanding Pointers in C*, BPB publications, 4th Edition, 2008
- [3] Kamthane Ashok, *Programming with ANSI and Turbo C*, Pearson Education,2006

MCA132 Web Technologies**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To help the students to understand the concept of HTML, CSS, Java script and PHP.

Learning Outcome

The student will be able to completely develop a dynamic website with data base backend.

Unit I.**(12)**

Fundamentals of Web: Internet, WWW, Web Browsers, and Web Servers, URLs, MIME, HTTP, Security.

HTML and CSS

HTML – XHTML – HTML 5, Creating simple web page, Basic text formatting, presentation elements, Phrase elements, Lists, Font, grouping elements, Basic Links, Internal document links, email link, Image, Audio and Video, image maps, image formats, Adding flash content and video, Tables – attributes, nested tables, Forms – Attributes, form controls, Frames- Frame set, nested frames, attributes. Introduction to HTML 5 - New tags of HTML 5 – embedding Media content, building input forms, painting on canvas.

Cascading Style Sheet

Introduction, What are CSS, Levels of Style sheet and specification formats, embedded style sheet, External style sheet, inline style sheet, classes, Class and ID method, DIV and SPAN tags. Inheritance with CSS. Introduction to CSS 3, HTML 5 and CSS3.

Unit II.**(12)****JavaScript**

JavaScript Implementation, JavaScript in HTML, Language Basics – Variables, operators, statements, functions, Data type conversions, reference types, Document object Model - browser object model - window object, location object, navigator object, screen object, history object, Events and Event handling, Button elements, Navigator object, validations with regular expressions. Introduction to Dynamic documents, Positioning elements, moving elements, elements visibility, changing colors and fonts, dynamic content, Locating mouse cursor, reacting to a mouse click, dragging and dropping of elements. Basic Animation with image using JavaScript.

Unit III.**(12)****PHP**

Introduction to Server side Programming, Introduction to PHP , PHP and HTML, essentials of PHP, Why Use PHP, Installation of Web Server,WAMP Configurations, Writing simple PHP program, embedding with HTML, comments in PHP, Variables, Naming Conventions, Strings, String Concatenation, String functions, float functions, Arrays, Array – Key pair value, Array functions, is SET, UNSET, gettype(), settype(), control statements (if, switch), Loops, User Defined Functions (with argument, return values), global variable, default value, GET -

POST method, URL encoding, HTML Encoding, Cookies, Sessions, Include statement. File : read and write from the file.

Unit IV.**(12)****MySql**

Introduction to MySQL, CRUD - Select statements, Creating Database/Tables, Inserting values, updating and Deleting, PHP with MySQL, Creating Connection, Selecting Database, Perform Database (query), Use returned data, close connections, file handling in PHP – reading and writing from and to FILE. Using MySQL from PHP (Building a Guestbook).

Unit V.**(12)****Object Oriented Programming with PHP**

Introduction to OOPS, creating classes, creating objects, setting access to properties and methods. Constructors, destructors, overloading and overriding of methods. Accessing PHP and HTTP Data. Reading POST and GET variables. Form validation.

Text Books:

- [1] Jon Duckett , *Beginning HTML , XHTML , CSS, and JavaScript*, Wiley Publishing , 2010
- [2] Steve Suehring, *JavaScript –Step by Step*, PHI, 2nd Edition, 2012
- [3] Matt Doyle , *Beginning PHP 5.3*, Willey Publishing, 2010

Reference Books:

- [1] Faithe Wempen, *HTML 5 Step by Step*, Microsoft Press, PHI, 2012.
- [2] David Sawyer McFarland , *CSS – The Missing Manual* , Pogue Press , O'Reilley Willey Publishing , 2008.

MCA133 Digital Logic**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To help students to understand the concept of number system, Boolean algebra, combinational & sequential logic circuits, and the concept of memory structure.

Learning Outcome

Upon successful completion of the course, students should be able to:

- Convert values from one number system to another number system, apply arithmetic operations to any number system, convert signed numbers to complementary system
- Write Boolean equations from truth tables in SOP or POS form, implement Boolean equations with logic gates, simplify Boolean expressions using Boolean Algebra and Karnaugh Map
- Design and understand the function of basic combinational logic circuits such as adder, subtractor, encoder, decoder, multiplexer and demultiplexer.
- Design and analyze sequential logic circuits such as latches and flip-flop, use flip-flops in designing sequential logic circuits and counters

Unit I. (14)**Digital Computer and Information**

Digital Computers, Number Systems, Arithmetic Operations, Decimal Codes, Alphanumeric Codes.

Combinational Logic Circuits

Binary Logic and Gates, Boolean algebra, DeMorgan's theorem, Simplification using Boolean laws, Standard forms, Karnaugh Map, Map Simplification (SOP and POS method), NAND and NOR Gates, Exclusive-OR Gates, Integrated Circuits.

Unit II. (12)**Combinational Logic Design**

Combinational Circuits, Design Topics, Analysis Procedure, Design Procedure, Decoders, Seven segment decoder, Encoders, Multiplexers, Binary adders, Binary Subtractor, Binary adder – subtractors, Binary Multipliers, Decimal Arithmetic.

Unit III. (11)**Sequential Circuits (FF's with Timing Diagram)**

Sequential Circuit Definitions, Latches, Clock, Types of Clock, positive, Negative edge triggered, Flip-Flops- SR, D, JK, Edge Triggered, T Flip-Flop, Master-Slave, JK Flip-Flop.

Unit IV. (11)**Registers and Counters**

Definition of Register and Counter, Registers, Shift Registers – Serial Transfer, Serial Addition, Shift register with Parallel Load and Bidirectional Shift Register, Synchronous Ripple Counter, Asynchronous, Synchronous Binary Counters, BCD counter, Up/Down counter.

Unit V.**(12)****Memory and Programmable Logic Devices**

Definitions, Random-Access memory, RAM Integrated Circuits, Array of RAM Ic's, Programmable Logic Technologies, ROM, Programmable Logic Array, Programmable Array Logic Devices, VLSI Programmable Logic Devices.

Text Book:

- [1] Mano, Morris M and Kime Charles R. *Logic and Computer Design Fundamentals*, Pearson education, 2nd edition, 2010.

Reference Books:

- [1] Tokheim, *Digital Electronics Principles and Applications*, Tata Mc Graw-Hill, 6th edition, 2009.
- [2] Malvino, Paul Albert and Leach, Donald P. *Digital Principles and Applications*, Tata Mc Graw-Hill, 4th edition, 2010.
- [3] Bartee, Thomas C. *Digital Computer Fundamentals*, Tata Mc Graw-Hill, 6th edition, 2008.

MCA134 Software Engineering**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

- To provide the students to understand the concepts software engineering. To prepare the students to develop the skills necessary to handle software projects. To make the students aware of the importance of software engineering principles in designing software projects.

Learning Outcome

On completion of the course the student will:

- Understand the importance of the stages in the software life cycle.
- Understand the various process models.
- Be able to design software by applying the software engineering principles.
- Understand the importance of Software quality and testing.
- Develop the quality of efficient project management.

Unit I.**(12)****Software and Software Engineering**

Nature of software- Defining software, Software Application Domains, Legacy Software, Software Engineering, The software process, Software Engineering practice – The essence of Practice, General Principles, Software Myths.

Process models

A generic process model – Defining a framework activity, identifying a Task Set, Process Patterns, Process Assessment and improvement, Prescriptive Process Models – The waterfall Model, Incremental Model, Evolutionary Process Model, Concurrent Model, Component based Development, The formal Methods Model .

Understanding Requirements

Requirements Engineering, Establishing the groundwork – Identifying Stakeholders, Recognizing multiple viewpoints, Working toward Collaboration, Asking the first questions, Eliciting requirements – Collaborative requirement gathering, Quality function Deployment, Usage Scenario Elicitation Work Products, Developing use cases, building the requirements model – Elements of the requirements Model, Analysis pattern, Negotiating requirements, validating requirements.

Case Study on requirement gathering based on some domain**Unit II.****(12)****Design Concepts**

The design within the context of Software Engineering, The design process – Software quality guidelines and attributes, The evolution of software design, Design concepts – Abstraction, Architecture, Patterns, Separation of concerns, Modularity, information hiding, Functional Independence, refinement, Aspects, Refactoring, Object Oriented design concepts Design classes, The design Model – Data Design elements, Architectural Design elements, Interface Design Elements, Component-Level Design elements, Deployment level Design elements.

Architectural Design

Software architecture – What is architecture, Why is Architecture important, Architectural descriptions, Architectural Decisions, Architectural style – Brief taxonomy of Architectural styles, Architectural Patterns, Organization and refinement, Architectural Design – Representing the system in context, Defining Archetypes, Refining the Architecture into components, Describing Instantiations of the system, Architectural mapping using Data flow – Transform Mapping, Refining Architectural Design.

Case study on architectural design**Unit III.****(12)****Component Level Design**

What is a component – An Object-Oriented View, The Traditional View, A Process-Related View, Designing class based components – Basic Design Principles, Component-level Design guidelines, Cohesion, Coupling, Component Design for WebApps – Content design at the Component level, Functional design at the Component level, designing traditional components – Graphical design notation, Tabular Design Notation, Program Design Language, Component based development- Domain Engineering, Component qualification, Adaptation, and Composition, Analysis and Design for reuse, classifying and retrieving components.

User Interface Design

The golden rules- Place the User in Control, Reduce the User's Memory load, Make the interface Consistent, Interface Analysis and Design models, The Process, Interface Analysis User Analysis, Task Analysis, Analysis of Display Content, Analysis of the Work Environment, Interface design steps – Applying Interface Design steps, User Interface design patterns, Design Issues, Webapp Interface design – Interface Design Principles and Guidelines, Interface Design workflow for WebApps.

Case study on UI design**Unit IV.****(12)****Quality Management**

Software Quality, Garvin's Quality Dimensions, McCall's Quality Factors, ISO 9126 Quality Factors, Targeted Quality factors, Transition to a Quantitative view, Achieving software quality- Software Engineering Methods, Project Management Techniques, Quality Control, Quality Assurance.

Testing Conventional Applications

Software testing fundamentals, internal and external view of testing, White-box testing, Basic path testing - Flow graph notation, Independent program path, Deriving test cases, Graph matrices-, , control structure testing – Condition testing, Data flow testing, loop testing-, Black-box testing – Graph- based Testing Methods, Equivalence Partitioning, Boundary Value Analysis, Orthogonal Array Testing, Model Based Testing, Testing for specialized environments, Architectures, and Applications – Testing GUIs, Testing of Client-Server Architectures, Testing Documentation and Help facilities, testing for Real-Time Systems, Patterns for software testing.

Testing Web Applications

Testing concepts for WebApps- Dimensions of Quality, Errors within a WebApp Environment, Testing Strategy, Test planning-, The testing process, Content testing- objective, Database Testing-, User Interface testing – interface testing strategy, Testing Interface

Mechanisms, Testing interface Semantics, Usability Tests, Compatibility tests, Component-level testing, Navigation testing-Testing navigation syntax, Testing navigation semantics-, Configuration testing- Server side issues, Client side issues-, Security testing, Performance testing-objectives, Load testing, Stress testing.

Concepts and Terminology

ISO 9000, SQA, Cost impact of software defects, Review metrics and their use, Formal technical reviews.

Case study on test cases

Unit V.

(12)

Process and Project Metrics

The management spectrum- The people, The product, The Process, The project-, Metrics in the process and project domains-Process metrics and Software Process improvement Project Metrics-, software measurement-Size Oriented metrics, Function Oriented Metrics, Reconciling LOC and FPMetrics, Object Oriented Metrics, Use case oriented metrics, WebApp project metrics-, Metrics for software quality – Measuring quality, Defect removal Efficiency.

Estimation for Software Projects

Observations on estimation, The project planning process, Software scope and Feasibility, Resources-Human resources, reusable software resources, Environmental resources, software project estimation, Decomposition techniques – Software sizing, Problem based estimation, Example of LOC based estimation, Example of FP based estimation, Process based estimation, Example of process based estimation, estimation with use cases, example of use case based estimation, Reconciling estimates, Empirical estimation models – The structure of Estimation model, COCOMO II Model, Software equation.

Project Scheduling

Project scheduling- Basic principles, The relationship between People and Effort, Effort Distribution, Scheduling – Time line Charts, Tracking the schedule, Tracking progress for an OO Project, Scheduling for WebApp projects.

Risk Management

Software risks, Risk identification- Assessing overall project risk, Risk components and drivers-, Risk projection – Developing a risk Table, Assessing Risk Impact, Risk Refinement, Risk Mitigation, Monitoring, and Management-, The RMMM plan.

Text Book:

[1] Pressman S Roger, *Software Engineering A Practitioner's Approach*, Mc Graw Hill, 7th edition, 2010

Reference Book:

[1] Sommerville, Ian, *Software Engineering*, Addison Wesley, 9th edition, 2010

Guidelines for case studies

1. The respective teacher in charge can decide the domain for the case studies. Either same domain for all the case studies or different domain for different case studies.
2. Software requirements specification case study needs to be aligned with the specification template.
3. Case study on architectural design expect detailed Architectural Context Diagram.
4. Case study on UI design need to target a Web Application Scenario.

MCA135 Probability and Statistics**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To help the students to understand & analyze data using suitable statistical tools.

Learning Outcome

The successful completion of this course will enable the students to understand the following concepts:

- Descriptive statistics.
- Concepts of probability.
- Formulation & Testing of hypotheses using suitable test statistics.
- Measures of central tendency

Unit I.**(12)**

Measures of central tendency- Arithmetic mean , Median and Mode. Partition values-quartiles,deciles and percentiles. Measures of dispersion – range, quartile deviation, standard deviation and coefficient of variation for grouped and ungrouped data. Skewness – Karl Pearson and Bowley’s measure of skewness. Correlation – Karl Pearson and Spearman’s correlation coefficient. Regression – Simple linear regression.

Unit II.**(12)**

Random experiment, sample space and events. Definitions of probability. Addition and multiplication rules of probability. Conditional probability and Bayes theorem. Random variables – Discrete and continuous. (univariate data) Probability mass functions and probability density functions. Expectation and variance.

Unit III.**(06)**

Probability distributions – binomial , Poisson and normal distributions. Approximation of binomial to Poisson distribution. Approximation of binomial and Poisson to normal distribution.

Unit IV.**(05)**

Concepts of statistic, parameter, sampling distribution and standard error. Chi square, t and F distributions.

Unit V.**(15)**

Statistical hypotheses-Simple and composite, Statistical tests, Critical region, Errors of Type I and Type II, Interval estimation – single mean and difference between two means (known and unknown variance), single proportion and difference between two proportions, sample size determination. Testing of hypothesis – null and alternative hypothesis, level of significance, Type I and Type II errors. Test for single mean and difference between means (known and

unknown variances), Paired t-test, Test for single proportion and difference between two proportions. Single variance and Ratio of two variances.

Unit VI. . (10)
Analysis of one-way and two-way classified data.

Text Book:

[1] Gupta S.C & Kapoor V.K , *Fundamentals of Mathematical statistics* , Sultanchand & sons, 2009.

Reference Books:

[1] Douglas C Montgomery, George C Runger, *Applied Statistics and Probability for Engineers*, Wiley student edition, 2004.

[2] Freund J.E, *Mathematical statistics*, Prentice hall, 2001.

[3] Berenson V Levine, *Basic Business Statistics*, Prentice-Hall India, 6th edition, 1996.

Question paper pattern

Part A

Consists 12 questions of 2 marks each, of which 10 have to be answered. The questions should cover the entire syllabus.

Definitions, statements, small problems with short answers to be asked in this section.

Part B

One question each from Unit I, Unit 2 and Unit 3. Each question carries 20 marks and can have a maximum of 4 sub questions. The student has to answer any two main questions.

Part C

One question each from unit 4, and Two question from unit 5. Each question carries 20 marks and can have a maximum of 4 sub questions. The student has to answer any two main questions.

MCA136 Human Resource Management**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To familiarize students with the concepts of HRM with respect to IT industry in specific, facilitate students in designing the recruitment and selection process with the support of IT. To impart knowledge on the important upcoming areas of HRM. To introduce the students the relevance of HRM in globalised and techno based economy.

Learning Outcome

- Students will learn to design the E-recruitment and E-selection process.
- Students will learn to prepare online training and development modules for specific organizations.
- Students will learn role and importance of IT in HR department.
- Students will learn the role of trade unions and employee engagement in the modern organizations.

Unit I. (07)**Human Resource Management**

Concept, Objectives, Scope, Functions and Models of HRM. Corporate Strategies and HRM.

Human Resource Management in Changing Environment

Human Resource Management in India, Paradigm Shifts in People Management, Problems and Challenges of Human Resource Management in India.

Importance of Strategic HRM in competitive driven economies. Exit policy and practices.

Scope of HR Accounting in modern organizations.

Unit II. (08)

Job Analysis: Job Description and Job Specification.

Human Resource Planning

Definition, Objectives, Scope and Importance, Methods of Forecasting,

Unit III. (12)**Talent Acquisition**

Recruitment: Importance and Sources of Recruitment

Selection: Importance and Process of Selection. Tests and Interviews for attracting and retaining the best talent.

Placement and Induction Process.

Performance Management

Meaning, Objectives, Scope and Purpose, Appraisal Process, Methods for Evaluating Performance, Problems and Challenges in Appraisal.

Unit IV. (12)**Human Resource Development**

Meaning, Objectives and Scope of Human Resource Development in India. Methods for Training workers and managers, Problems and Challenges of training and Development in India, Evaluation of Training Effectiveness.

Career Planning and Development

Career, Career Planning, Need for Career Planning, Process of career planning and development. Organizational and Individual career planning, succession planning.

Internal mobility and external Mobility

Importance and types of internal mobility. Meaning the types of external mobility.

Unit V. (08)**Reward Management**

Job Evaluation: Introduction, meaning and types of job evaluation

Role of reward system. Definition and Objectives, Theory of Wages, Components of worker compensation, Components of executive compensation. Problems and Challenges in promoting equity in compensation and reward systems.

Fringe benefits of top 10 multi national companies.

Unit IV. (12)**Labor Management Relations**

Definition, Objectives, Features of Industrial Relations in India, Methods of Managing Employment Relationship.

Trade Unions

Leadership: Meaning, importance and Types of Leaders. Leaders vs. Managers. Definition, Objectives and Purpose of Trade Unions, Trade Union Movement in India, Trade Union At 1926, Issues, Problems and Challenges of Trade Union in India.

Collective Bargaining

Definition, Objectives and Scope of Collective Bargaining, Process of Collective Bargaining, Types of Collective Bargaining, Collective Bargaining in India, Productivity Bargaining.

Workers Participation in Management

Definition, Objectives and Scope of Workers Participation in Management, Levels of Participation, Workers Participation in India.

Text Book:

[1] P.Subba Rao, *Essential of HRM and IR, Text and Cases, Himalaya Publications*, 7th Edition, 2011.

Reference Books:

- [1] H. John Barnardian & Jyoce E.A. Russel, *Human Resource Management and Experimental Approach*, McGraw Hill, 6th Edition, 2010.
- [2] David A. Decezo & Stephen P. Robbins, *Personnel/ Human Resource Management*, Prentice Hall India, 7th Edition, 2009.
- [3] Aswathappa, *Human Resource Management*, Tata McGraw Hill, 10rd Edition, 2011.
- Edwin B Flippo, *Human Resource Management*, Tata McGraw Hill, 10th Edition, 2011.
- [4] William B. Werther & Keith Davis, *Human Resource and Personnel Management*, McGraw Hill, 7th Edition, 2010.

Question Paper Pattern

Question paper has to be set for total marks of 100.

Part–A: Five questions to be answered out of seven $2 \times 5 = 10$

Part–B: Five questions to be answered out of six $5 \times 5 = 25$

Part–C: Three questions to be answered out of four $15 \times 3 = 45$

Part–D: Case study (compulsory) $20 \times 1 = 20$

Total Marks = 100

MCA151 C Programming Lab*Total Hours/Semester: 60**No of Hours/Week: 04****All the programs should be based on a unique domain.*****Section A**

1. Implementation of the various Data Types with modifiers and type conversion in C.
2. Demonstration of nested if and switch... case structure
3. Implementation of various Control structures in C
4. Implementation of arrays
5. Implementation of multidimensional arrays
6. Implementation of functions :call by value, call by reference, passing of arrays, Recursion
7. Demonstration of various user defined string operations
8. Implementation of the storage types
9. Demonstration of pointer operations.
10. Demonstration of macro processing.

Section B

11. Implementation of structures and array of structures
12. Implementation of Union.
13. Implementation of pointers to structures and unions.
14. Demonstration of dynamic allocation of memory
15. Demonstration of bitwise operations.
16. Demonstration of various Text file operations.
17. Demonstration of various fixed shapes with some animation
18. Demonstration of different graphics functions

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA152 Web Technologies Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

1. Create a Web page by making use of the following tags : Headers, Linking and Images.
2. Create a Web page that will have the following: Frames, Unordered Lists, Nested and ordered Lists
3. Create a Web page Layout with Tables and all its attributes
4. Create a Web page that will have Application form (Forms) , make use of Image Maps and <meta> Tags
5. Create an External Style Sheet that defines the style for the following tag : H1, H2, Body , P, Li .
6. Create an Internal Style Sheet that defines a style for Positioning elements & setting the background (color / image)
7. Create a Style Sheets that defines the style with class method , Id method , make use of DIV and Span TAG
8. Create a style Sheet that demonstrate Box Model
9. Write a JavaScript program to Demonstrate the use of Variable , message box , and loops
10. Write a JavaScript Program to demonstrate Functions (predefined / user defined)
11. Write a JavaScript program to demonstrate Event Handling
12. Object Creation and modification in JavaScript
13. Write a PHP program to demonstrate GET and POST method of passing the data between pages
14. Write a PHP program to demonstrate Array , Key-pair values
15. Write a PHP program to read and write the Data from the Database
16. Create a PHP page that uses Session and cookies.
17. File Handling in PHP
18. Implementing the OOPs concept in PHP

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA231 Microprocessors and Interfacing Techniques**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To help students to understand the basics of 8085 microprocessor-based systems and assembly language programming. This course also gives the introduction to 8051 microcontroller.

Learning outcome:

Upon successful completion of the course, students should be able to:

- Identify the basic element and functions of microprocessor.
- Describe the architecture of microprocessor and its peripheral devices.
- Demonstrate fundamental understanding on the operation between the microprocessor and its interfacing devices.
- Apply the programming techniques in developing the assembly language program for microprocessor application.
- Understand the basic concept of microcontroller.

Unit I. (12)**Microprocessor 8085**

Introduction to Microprocessor 8085 –Signals -Address Bus, Data Bus, Control & status signals, Power supply and Frequency signals, Externally initiated signals, serial I/O ports

8085 Machine cycles and bus Timings

Opcode Fetch Machine cycle, Memory Read, Memory Write, I/O Read and I/O Write Machine cycles, Calculation of execution time for a program with examples

Unit II. (11)**Architecture of 8085 MPU**

Block Diagrams, Registers, Flags, ALU, Timing and Control Unit, Instruction Decoder, Serial I/O Control, Stack, PC, Address/Data Buffers

Unit III. (12)**Introduction to 8085 programming**

The 8085 programming model, Instruction Classification, Data Format and storage, 8085 instruction Set Addressing Modes, Data Transfer Operations, Arithmetic Operations, Logic Operations, Branch Operations, Programming Techniques, Writing simple programs.

Unit IV. (14)**Programming Techniques with Additional instructions:**

Looping Counting and indexing Additional data transfer and 16 bit Arithmetic Instructions, Arithmetic operations related to memory, Logic operations: Rotate, Compare. Writing assembly language programs- Binary and BCD addition of two 32 bit numbers, Binary and BCD subtraction of 16 bit number, Multiplication and division of 8 bit numbers, shifting 8 bit number by 1or 2 bit etc.,.

Counters and Time Delays:

Counters and Time delays, Illustrative program, modulo Ten counter, Subroutine concepts, Subroutine call and return instruction

Interrupts

Introduction – INTR, TRAP, RST 7.5, 6.5, 5.5 – RST, SIM and RIM instructions

Unit V.**(11)****8255A**

Programmable peripheral interface – Block Diagram – Control Logic, Control Word – Modes of operations with examples, Mode 0, Mode 1, BSR Mode, Control word for each modes of operation Programming in 8255A with an example.

Text Books:

- [1] Ramesh.S.Goankar ,*Microprocessor Architecture, Programming & Applications With 8085*, 5th Edition – Penram International – 2012. ISBN 81-87972-09-2.

Reference Books:

- [1] Hall.D.V., *Microprocessor and Digital System*, McGraw Hill Publishing Company, 2nd Edition, 2008.
[2] Charles M Gilmore, Pal Ajit, *Microprocessor Principles and Applications*, Tata McGraw Hill, 2nd Edition, 2009.

MCA232 Object Oriented Programming using C++**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

- To demonstrate the usage of data abstraction, encapsulation, inheritance.
- Learn the other features of C++ language including templates, exceptions, STL
- To master the techniques of software development in the C++ programming language and demonstrate the techniques by the solution of a variety of problems spanning the breadth of the language.

Learning outcome

- Master the principles of object oriented programming in well written modular code
- Demonstrate significant experience with the program development environment.

Unit I.**(12)****OOP Paradigm**

Evolution of programming methodologies, Origins of C++, Procedural Approach Vs Object-Oriented Approach. Principles of OOP: Encapsulation and Abstraction, Message Passing, Inheritance – Reusability, Extensibility, Polymorphism – Overloading, Dynamic Binding.

Comparison of C and C++

Limitations of C, Introduction to C++, Structure of the C++ program, Added features of C++ over C – Storage Classes, Reference variables, Inline functions. Simple I/O using cin & cout, I/O formatting and I/O functions. .

Introduction to Objects and Classes

Defining the class, Defining Data members and member functions, Creating Objects of Class, Access Specifiers – private, public and protected. Scope Resolution Operator, Nested Classes, Local Classes. Friend Functions and Friend Classes, passing objects as functions-returning objects – Static Members. this pointer, returning values using this pointer. Comparison of class with structure.

Unit II.**(12)****Constructors and Destructors**

Purpose of Constructors and Destructors, Default Constructors, Constructors with & without parameters, Constructor Overloading, Copy Constructor-Deep copy and shallow copy. Invoking Constructors and Destructors.

Pointers in C++

Introduction-Pointer variable declarations and Initialization- Pointer Operators-Passing Arguments to Functions by Reference with pointers. Using const with Pointers-Pointer Expressions and Pointer Arithmetic-Relationship between pointer and Arrays-Array of Pointers-Function Pointers. Array of Pointers to String, memory management – new and delete, pointer to object – referencing members using pointers, wild pointers, Dangling pointers, Smart pointers.

Unit III. (12)**Polymorphism**

Overloading Concepts Function Overloading: Functions with different sets of parameters, default and constant parameters. Operator Overloading: Defining Operator Function, Rules for overloading Operators. Creating prefix and postfix forms of the increment and decrement Operators- operator Overloading using a Friend Function-Overloading new and delete-overloading some special operators [],(),->,overloading the comma operator.

Type Conversions:

Basic to Class, Class to Basic and one Class to another Class type.

Unit IV. (12)**Inheritance**

Basic Concepts, Reusability & Extensibility. Defining derived classes, protected access specifier in Base class – public, private & protected inheritance – constructors and destructors in derived classes – Types of Inheritances-Single, Multiple, Multilevel, Hierarchical, Hybrid- Ambiguity in multiple inheritance- Virtual base class, Virtual destructor.

Virtual Functions

Normal member functions accessed with pointers, virtual member function access, late binding, pure virtual function, abstract classes.

Templates

Introduction to Generic Functions- A generic swap function, Overloading a Function Template. Generic Classes – A stack generic class, type name and template keywords, Template Restrictions, The power of Templates.

Unit V. (12)**Streams**

Classic streams Vs Standard Streams-Stream Input-Stream Output-Unformatted I/O using read, write. Introduction to Stream Manipulators-Stream Form States and Stream Manipulators, Stream Error States

File Processing

Introduction-Data Hierarchy-Files and Streams-Creating a Sequential file-Reading a data from a Sequential File-Updating Sequential files-Random Access Files-Creating a Random Access file-Writing data Randomly to a Random-Access File-Reading from a Random Access File Sequentially. **Exception Handling** Introduction- Exception Handling overview, When to use Exception Handling, Catching Class Types, Using Multiple catch statements, Catching All Exception, Restricting Exception, Re throwing an exception, throw statement, Uncaught exception,, and Built-In Exceptions. Introduction to Standard Template Library– Introduction to Containers- Iterators.

Text Book:

[1] Deitel & Deitel, *C++ How to program*, Pearson Education Asia, 6th Edition, 2010.

Reference Books:

[1] Schildt Herbert, *The Complete Reference C++*, Tata McGraw Hill, 4th Edition, 23rd reprint, 2009.

[2] M T Somashekar, D S Guru, *Object Oriented Programming with C++*, PHI, 2nd Edition , 2012

MCA233 Operating Systems**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To acquire the fundamental knowledge of the operating system architecture and components.
To understand its behavior and performance while handling various tasks with respect to different system architecture.

Learning Outcome

Upon completion of the course students will be able to:

- Describe how operating systems have evolved over time and its working process.
- Brief all the tasks performed by the operating systems.
- Understand the internal structure of the operating system with relevant system call or functions.
- Solve problems based on process and memory management.
- Identify potential threats to operating systems and find scope to perform further study in the security features and the relevant design.
- Describe how issues are influencing operating system design through assignments on latest developments.

Unit I.**(10)****Overview**

Introduction: Operating system definition, Computer system organization, structure, architecture and operations, process and storage management, Protection and security, Distributed systems, Special purpose systems, Computing Environments, Open-source Operating Systems.

System structure: operating system services, user interface, system calls, system programs, OS design, Implementation and structure, virtual machines, system boot.

Unit II.**(13)****Process Management**

Process: Process concepts, scheduling, operations on processes, Inter process communication, Examples of IPC systems, Communication in client server systems, Threads, Multi threading models, threading issues.

Scheduling: Basic concepts, scheduling criteria, scheduling algorithms, Thread scheduling, Multiple-processor scheduling.

Unit III.**(13)****Process Coordination**

Synchronization: Critical section problems, Peterson solution, Introduction to semaphores, classic problems of synchronization, Monitors, synchronization examples, atomic transaction.

Deadlock: System model, deadlock characterization, methods for handling deadlock, deadlock prevention, avoidance, detection and recovery from deadlock.

Unit IV. (12)**Memory Management**

Memory Management Strategies: Background, swapping, Memory allocation, Paging, Structure of the page table, Segmentation, Example: the Intel Pentium.

Virtual Memory Management: Demand paging, Page replacement, allocation of frames, thrashing, memory mapped files, Allocating kernel memory.

Unit V. (12)**Storage Management and Case Study**

File system: File concepts, access methods, directory and disk structure, File system mounting, File sharing, Protection, directory implementation, allocation methods, free-space management.

I/O Systems: I/O hardware, Application I/O Interface, Kernel I/O subsystem, Transforming I/O requests to hardware operations.

Case study: Windows XP: History, Design principles, system components, environmental subsystems, file systems.

Textbook:

1. Silberschatz, P.B. Galvin, G. Gadne, *Operating System Concepts*, Wiley-India Edition, 8th Edition. 2011.

Reference Books:

1. William stallings, *Operating system – Internals and Design Principles*, 7th Edition, Prentice Hall, 2010
2. Elmasri, E., Carrick, A.G. and Levine, D, *Operating Systems: A Spiral Approach*, McGraw Hill, 2010.
3. McHoes, A.M. and Flynn, I.M., *Understanding Operating Systems*, 6th Edition, Thomson, 2011.
4. Dhamdhere, D.M., *Operating Systems: A Concept-based Approach*, 2nd Edition, McGraw Hill, 2006.
5. Dietel D, *Operating System*, 3rd Edition, Pearson Education, 2004.

MCA234 Relational Database Management System**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To provide strong foundation for database application design and development by introducing fundamentals of database technology.

Learning Outcome

- Understanding the fundamentals of RDBMS.
- Understanding the database design process and its significance.
- Logic development for database application programming.
- Insights into recent developments in database technologies.

Unit I.**(12)****Introduction to Database system concepts, file structures and conceptual modeling**

Database system concepts and architecture – Data models, schemas and instances, DBMS architecture and data independence, Database languages and interfaces, database system environment, Classification of DBMS.

Disk storage, basic file structures and hashing

Secondary storage devices, buffering of blocks, Placing File Records on Disk Operations on Files, Files of Unordered Records, Files of Ordered Records hashing techniques.

Data modeling using ER model

Entities, attributes and relationships, Different types of attributes, E- R Diagrams, Specialization and generalization, constraints and characteristics of specialization and generalization, Relationship types of degree higher than two.

Unit II.**(12)****Relational Data Model and Database design**

Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations.

ER and EER to Relational Mapping

Relational database design using ER to Relational Mapping, Mapping EER Model concepts to relations.

Database Design

Informal design guidelines for Relation schemes, Functional dependencies, Normal forms based on primary keys, General definitions of second and third normal forms.

Unit III.**(12)****Advanced normalization concepts and SQL**

Boyce – Code normal form, multi-valued dependencies and fourth normal form, Join dependencies and fifth normal form.

Basic SQL

SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT, DELETE, and UPDATE Statements in SQL, Additional features of SQL.

Unit IV.**(11)****Advanced SQL and Transaction Management****Complex Queries, Triggers, Views, and Schema Modification**

More Complex SQL Retrieval Queries, Specifying Constraints as Assertions and Actions as Triggers, Views (Virtual Tables) in SQL, Schema Change Statements in SQL.

Transaction Management

Transaction - Introduction to transaction processing, transaction and system concept, Desirable properties of transaction, Transaction support in SQL, concurrency control techniques – Two phase Locking techniques for concurrency, timestamp based protocol.

Unit V.**(13)****Overview of Distributed database, object, object relational and XML database****Distributed Database**

Introduction to Distributed database concepts, Types of Distributed Database Systems, Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design, Overview of Transaction Management in Distributed Database, Overview of Concurrency Control and Recovery in Distributed Database.

Object, object relational and XML database

Object and Object-Relational Database– Overview of Object Database Concepts, Object-Relational Features: Object Database Extensions to SQL, The ODMG Object Model and the Object Definition Language ODL, The Object Query Language OQL.

XML Databases

A data model for XML, Querying XML data, Efficient evaluation of XML queries.

Text Book:

- [1] Elmasri & Navathe, *Fundamentals of Database Systems*, Addison-Wesley, 6th Edition, 2010.

Reference Books:

- [1] Korth F. Henry and Silberschatz Abraham, *Database System Concepts*, McGraw Hill, 6th Edition, 2010.
 [2] O'neil Patric, O'neil Elizabeth, *Database Principles, Programming and Performance*, Argon Kaufmann Publishers, 2nd Edition, 2002.
 [3] Ramakrishnan and Gehrke, *Database Management System*, McGraw-Hill, 3rd Edition, 2003.

MCA235 Discrete Mathematical Structures**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To prepare the students for a background in abstraction, notation, and critical thinking in the Discrete Mathematics closely related to computer science.

Learning Outcome

The successful completion of this course will enable the students to :

- Construct mathematical arguments using logical connectives and quantifiers.
- Verify the correctness of an argument using propositional and predicate logic and truth tables.
- Understand how Graphs are used as tools and Mathematical Models in the study of networks
- Construct proofs using direct proof, proof by contraposition, proof by contradiction, proof by cases, and mathematical induction.
- Apply algorithms and use definitions to solve problems to prove statements in elementary number theory.
- Perform operations on discrete structures such as sets, relations and functions and be familiar with concepts like Groups and Rings.

Unit I.**(15)****Foundations**

How to do Mathematics? – Compound statements – Existential and Universal statements – Proof techniques – Logical operations – Logical equivalence- Conditional statements – Universal and Existential quantifiers – Concept of a function – Types of functions – Composition of functions.

Unit II.**(15)****Techniques**

Introduction to numbers – Divisibility – Greatest common divisor – Existence and uniqueness of prime factorization – Partition of a set – Partition of a positive integer – Even and odd permutations – modular arithmetic – Latin squares.

Unit III.**(15)****Networks**

Types of relations – Graphs as network – Types of graphs-Representation of graphs – Representation of relations through graphs – Paths and Cycles- Eulerian and Hamiltonian properties of paths – Equality of graphs – Trees – Coloring of graphs – Max-Flow –Min-Cut theorem.

Unit IV.**(15)****Algebraic Structures**

Groups – Axiom of a group – Examples and basic algebra in groups – Order of an element of a group – Isomorphism of groups – Cyclic groups – Subgroups – Cosets and Lagrange's theorem – Rings-Fields.

Text Book:

- [1] N L Biggs, *Discrete Mathematics*, Oxford University Press, New Delhi, 2nd Edition, 2003.

Reference Books:

- [1] R. P. Grimaldi, *Discrete and Combinatorial Mathematics*, Pearson education, 5th Edition, 2004.
[2] B. Kolman, R. C. Busby and S. C. Ross, *Discrete Mathematical Structures*, Pearson Education, 5th Edition, 2004.
[3] T. Koshy, *Discrete Mathematics with Applications*, Elsevier Academic Press, London, 2004.
[4] K. H. Rosen, *Discrete Mathematics and Its Applications*, Tata McGraw-Hill, 6th Edition, 2006.
[5] G.S. Rao, *Discrete Mathematical Structures*, New Age International, 2009.
[6] J. P. Trembly and R. Manohar, *Discrete Mathematics with Applications to Computer Science*, Tata McGraw-Hill, 2003.

FORMAT OF QUESTION PAPER

Part	Unit and No. of subdivisions to be set in the unit		No. of subdivisions to be answered	Marks for each subdivision	Max. Marks for the part
A	Unit I	1	5	2	10
	Unit II	1			
	Unit III	2			
	Unit IV	1			
B	Unit I	3	10	5	50
	Unit II	3			
	Unit III	3			
	Unit IV	3			
C	Unit I	1	4	10	40
	Unit II	1			
	Unit III	1			
	Unit IV	1			
Total Marks					100

MCA236 Accounting and Financial Management

Total teaching Hours/Semester: 60

No of Lecture Hours/Week: 04

Objective

To develop knowledge of recording business transactions.

To develop skills in preparing financial statements

To develop skills in analysing financial statements

To equip upcoming programmers to identify and solve finance related problems and manage finance related projects.

Learning outcome

- Understand Accounting concepts and Terminology of accounting and finance
- Identifying and studying the asset, liability and Income and Expenditure Requirements and positions of the business
- Knowledge on cost review and preparation of Statement of cost and profit
- Understanding of forecast of the future cash, income and expense requirements.

Accounting

(02)

Basic terms -Principles- Concepts - Conventions- IFRS

Double Entry System of accounting

(10)

Journal, Ledger, Cash Book, Closing of Books of Accounts and Preparation of Trial Balance.

Final Accounts

(08)

Trading, Profit and loss Accounts and Balance Sheet of sole proprietary concern with normal closing and adjusting entries- Adjustments – Closing stock-Depreciation- Outstanding expenses-Prepaid expenses-Bad debts-provision for bad debt.

Final accounts of Joint Stock Companies

(04)

Profit and Loss Account- Profit and Loss Appropriation Account and Balance Sheet.

Financial Management

(02)

Meaning Role and Goals of Financial Management.(Theory only)

Fund Flow Statement

(08)

Meaning of the terms – Fund, flow of fund and working capital cycle. Preparation of Fund Flow Statement. Simple adjustments.

Ratio Analysis

(08)

Meaning advantages and Limitations. Types of ratios and their usefulness. Calculation of Current Ratio- Liquid Ratio- Cash ratio- Debtors Turnover Ratio- Creditors Turnover Ratio- Inventory Turnover Ratio- Working Capital Turnover Ratio- Gross Profit Ratio- Net profit Ratio- Operating Ratio- Operating Profit Ratio – Expense Ratio- Debt Equity Ratio – Fixed Asset Ratio- Earnings Per Share-Dividend per share- and their interpretations.

Costing (06)

Meaning, Nature and importance. Preparation of Cost Sheet.

Marginal Costing (04)

Meaning, Nature, scope and importance. Break-Even Analysis.

Budget & Budgetary Control (06)

Budget and Budgetary Control - Meaning and Importance. Different types of Budgets. Preparation of Flexible Budget and Cash Budget.

Introduction to Computerized Accounting System (02)

Coding Logic and Codes Required, Master File, Transaction Files, Introduction to Documents used for Data Collection, Processing of different files and outputs obtained, Application Packages in Accounting Tally.

Recommended book:

- [1] C. Mohan Juneja, *Fundamentals of Accounting and Financial Management*, Kalyani Publishers 2011.

Reference Books:

- [1] S.P. Jain and K.L Narang, *Advanced Accountancy*, Kalyani Publishers, 18th Edition, 2011.
[2] I M Pandey *Management Accounting*, Third revised Edition, 2010
[3] Lavy and Sarnat, *Principles of Financial Managament*, Prentice Hall.
[4] Arnolel, *Financial accounting*, PHI (Paper Back Edition).
[5] S N Maheshwari S K Maheshwari, *An Introduction to accountancy*, 10th Edition, 2010.
[6] Shashi K Gupta, R K Sharma, *Financial Management Theory and Practice*, 6th Revised Edition 2010.

Question Paper Pattern**Section A 10 x 2 = 20**

Answer any Ten questions out of twelve. Each question carries two marks.

Section B 4 x 10 = 40

Answer any four questions out of six. Each question carries ten marks.

Section C 2 x 20 = 40

Answer any two questions out of four. Each question carries 20 marks.

MCA251 Assembly Language Programming Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

Write assembly language programs for the following:

1. Write a program to add N one byte number.
2. Write a program to interchange N one bytes of data.
3. Write a program to check whether the 4th bit of a number is zero or one.
Display FF if 1 otherwise display 00.
4. Write a program to find the first 10 terms of a Fibonacci sequence
5. Write a program to find sum of first 10 terms of odd and even series.
6. Write a program to check whether a byte belongs to the 2-out-of-5 codes. Display FF if it is a 2-out-of-5 code otherwise 00. (Number is 2-out-of-5 code if the left most three bits are zero and in the remaining five bits there are exactly two 1's)
7. Write a program to perform linear search over a set of N numbers.
Display FF and its position if found otherwise 00.
8. Write a program to add two 32-bit binary numbers.
9. Write a program to add two 32-bit BCD numbers.
10. Write a program to subtract a 16-bit number from another 16-bit number.
11. Write a program to subtract a 16-bit BCD number from another 16-bit BCD number.
12. Write a program to multiply two 8-bit numbers.
13. Write a program to divide a 16-bit number by an 8-bit number.
14. Write a program to find the largest and smallest of N numbers.
15. Write a program to sort the numbers in ascending and in descending order using bubble sort.
16. Write a program to display a rolling message.
17. Write a program to determine the HCF of two one byte numbers.
18. Write a program to display FF and 00 alternatively with 1.5 sec delay.
19. Write a program to check whether a one byte number is a palindrome or not.
20. Write a program to prepare a look-up table for the squares of one-digit BCD numbers.
21. Write a program to simulate the throw of dice.
22. Write a program to determine the LCM of two one byte numbers.
23. Write a program to simulate a BCD counter to count from 0 to 100.
24. Write a program to simulate a stopwatch with a provision to stop the watch.

25. Write a program to implement block move with the without overlap condition.
26. Write a program to interface keyboard using 8255A interface.
27. Write a program to interface Seven Segment Display using 8255A interface.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA252 C++ Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

1. Implement Classes and Objects.
2. Implement Constructors and Destructors with array of Objects.
3. Implement Passing and returning parameters by reference.
4. Demonstrate Function Overloading.
5. Demonstrate overloading of different operators – incr & decr operators with post & pre forms, new, delete, [], () and arithmetic operators.
6. Implement pointer sort operation.
7. Demonstrate friend functions and friend classes.
8. Implement different types of inheritances like Multiple, Multilevel and Hybrid.
9. Demonstrate the use of Virtual Functions.
10. Demonstrate the use of abstract classes.
11. Demonstrate I/O streams and functions.
12. Overload << and >> operators as a member and as a non-member operator functions.
13. Create a file to store some records and search for a particular record and display it.
14. Demonstrate Namespaces and Volatile member functions.
15. Perform all possible Type Conversions.
16. Create function Templates and overload the function Templates.
17. Create a generic stack class and member functions to perform stack operations.
18. Implement Exception Handling with minimum 5 exception classes including two built-in exceptions.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA331 Data Structures**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

Data Structure is considered as one of the fundamental paper towards a more comprehensive understanding of programming and application development. Student is expected to work towards a sound theoretical understanding of Data Structures and also complement the same with hands on implementing experience

Learning Outcomes:

- Understand the need for Data Structures when building application
- Appreciate the need for optimized algorithm
- Able to walk through insert and delete for different data structures
- Ability to calculate and measure efficiency of code
- Appreciate some interesting algorithms like Huffman, Quick Sort, Shortest Path etc
- Able to walkthrough algorithm
- Improve programming skills

Unit I. (12)**Introduction and overview**

Introduction, Basic Terminology, Data Structures, Operations, Algorithms: Time & Space Complexity, Algorithmic Notation, Abstract Data Types.

Stacks and Queues

Stacks, Array Representation, Arithmetic Expressions, Polish Notation, Application of Stacks, Recursion, Towers of Hanoi, Implementation of Recursive procedures by Stack, Queues, Queue Array Representation .

Unit II. (11)**Linked Lists**

Introduction, Linked lists and Memory Representation, Traversing, Searching, Memory Allocation, Garbage Collection, Insertion, Deletion, Circular Linked list, Two-way Lists(Doubly). Linked List Implementation of Stack and Queue

Unit III. (12)**Sorting**

Introduction, Sorting, Insertion Sort, Selection Sort, Shell Sort, Merging, Merge-Sort, Quick Sort, Radix Sort, External Sorting

Searching

Hashing, Chaining, Linear Probe, Double Hashing, Text Searching using Knuth-Morris-Pratt algorithm, Regular Expression Matching

Unit IV.**(12)****Trees**

Introduction, Binary Trees, Representing Binary Trees in memory, Traversing Binary Trees, Traversal Algorithm using Stacks, Threads (Threaded Binary Trees), Binary Search Trees, Searching, Inserting and deleting in Binary Search Trees, Heap, Heap sort, Huffman's Algorithm.

Balanced Tree

AVL Trees: AVL Balance Factor, Balancing Trees, AVL node structure, AVL insert and delete algorithm walkthrough with examples. Other Balanced Tree Variations (Splay Trees and Red-Black Trees)

Unit V.**(13)****B-Trees**

B-Trees: B-Tree insertion, Deletion, Traversal and Search algorithm walkthrough with examples, Applications of B-Trees, Variations of BTree (B+Tree, B*Tree)

Graphs

Graph Theory Terminology, Sequential representation of Graphs, Adjacency matrix, Path matrix, Linked representation of a Graph, Operations on Graphs, Depth First and Breadth First Traversing a Graph, Shortest Path Algorithm, Minimum Spanning Tree Algorithm.

Text Books:

- [1] Gilberg, F Richard & Forouzan, A Behrouz, *Data Structures A Pseudocode approach with C*, 2nd Edition, Cengage, 2008.

Reference Books:

- [1] Horowitz Sahni Anderson-Freed, *Fundamental of Data Structures in C*, Universities Press, Reprint 2008.
[2] Richard Johnsonbaugh, *Algorithms*, Pearson Education, 2nd Edition, 2008
[3] Robert Sedgwick, *Algorithm in C++*, Addison-Wesley Publishing Company.
[4] Knuth, Donald E, *Art of Computer Programming, Sorting & Searching*, Addison-Wesley, 2005.

MCA332 UNIX Operating System

Total teaching Hours/Semester: 60

No of Lecture Hours/Week: 04

Objective

The course provides comprehensive understanding of the layered architecture of UNIX operating system, system calls, and file system structure. It also focuses on acquiring skills needed to develop UNIX shell programs, making effective use of wide range of UNIX programming standard and tools.

Learning outcome

Upon successful completion of the course the student would be able to

- Demonstrate a broad and integrated understanding of UNIX architecture
- Understand UNIX file system, process management, memory management and inter-process communication.
- Able to write shell scripts for basic and advanced level shell programming.
- Able to create programs with awk.

Unit I.

(12)

Introduction to UNIX

History, System structure, Users Perspective, OS Services. Architecture, System Concepts. The Buffer Cache: Headers, Structure of the Buffer Pool, Scenarios, Reading and writing Disk Blocks, Advantages and disadvantages of buffer cache. Algorithms: getblk, brelse, bread, breada, bwrite

File Systems

INODES, Structure of a regular file, Directories, Conversion of a path name to an INODE, Super Block, INODE assignment, Allocation of Disk Blocks, System calls for the file system: Open, Read, Write, Close, Pipes, Mounting and Unmounting Files. Algorithms: iget, iput, ialloc, ifree, open, read, write, creat.

Unit II.

(12)

UNIX shell environment

General purpose utilities, The File system, Handling Ordinary files, Basic File attributes, The Shell, The process, Hard links, Symbolic links, Umask, Modification and access time , Simple Filters: pr, head, tail, cut, paste, sort, uniq, tr, Filters using regular expressions: grep and sed ,Advanced Filters-awk, Essential System Administration

Unit III.

(12)

UNIX shell programming

Essential Shell Programming: read, using command line arguments, exit and exit status command, logical and conditional operators, if condition, using test and [], case, expr, Looping – while, for, set and shift, trap, debugging, functions.

Advanced Shell Programming: Shells, sub shells, export, running a script in current shell, eval, exec.

Unit IV.**(12)****Processes**

Process States and Transitions, Layout of System Memory, Context of a Process, Manipulation of the process address space, Process Control: Creation, Signals, Process termination, Awaiting process termination, invoking other programs, The Shell, System Boot and Init Process, Process Scheduling and Time: Process scheduling, System calls for time, Clock.

Algorithms: fork, exit, wait, exec

Unit V.**(12)****Memory management and The I/O sub system**

Swapping, Demand Paging, the I/O sub system: Driver Interfaces, Disk Drivers, Terminal Drivers, and Streams.

Inter process Communication

Process Tracing, System V IPC: Messages, Shared memory, Semaphore, Network Communications: Sockets.

Algorithms: msgsnd, msgrcv, shmat, semop,

Text Book:

[1] Bach M.J., "*The Design of the Unix Operating System*", Prentice Hall India, reprint 2009.

[2] Sumitabha Das, "*Unix Concepts and Applications*", Tata McGraw-Hill, Eighth reprint 2008.

Reference Books:

[1] Behrouz A.Forouzan, Richard F.Gilberg, "Unix and Shell Programming", CENERAGE Learning, seventh reprint 2009.

[2] Richard Stevens, "Advanced programming in the UNIX environment ", Addison Wesley, Edition reprint 2009.

MCA333 COMPUTER ARCHITECTURE**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objectives:**

To enable the students to learn the basic functions, principles and concepts of Computer architecture. This paper helps the students to learn the fundamental aspects of computer architecture and design. This paper focuses on processor design, control unit design techniques and IO interfacing.

Learning outcomes:

On successful completion of the course the students should have:

- Understood computer architecture
- Understood number systems, I/O, Registers and memory
- Understood processor design ,control unit design
- Understood IO interfacing

UNIT I.**(13)****Computer System**

Computer components – computer function – instruction fetch and execute – interrupts – I/O functions – interconnection structures – Bus interconnection - Bus structure – multiple bus hierarchies -elements of bus design

Memory:

Computer memory system overview – characteristics of memory system – memory hierarchy - cache memory principles – elements of cache design- cache size – mapping function – replacement algorithms – write policy – internal memory semiconductor memory – organization – DRAM and SRAM – types of ROM – chip logic – external memory- magnetic disk magnetic read write mechanisms – data organization and formatting – physical characteristics – disk performance parameters – RAID – optical memory

UNIT II.**(11)****Input/output organization:**

External devices – I/O modules – programmed I/O – interrupt driven I/O- DMA –I/O processor – interface circuits – serial port – parallel port – standard I/O interfaces – PCI bus , SCSI bus, USB bus

UNIT III.**(12)****Computer Arithmetic**

The arithmetic and logic unit – integer arithmetic – negation –addition - subtraction – multiplication and division –floating point representation – principles – IEEE standard for binary floating point representation – floating point arithmetic addition and subtraction – multiplication and division – precision considerations

UNIT IV**(12)****Central processing unit:**

Instruction sets characteristics – types of operands – types of operations – addressing modes - instruction formats- processor organization – register organization – instruction cycle – instruction pipelining- reduced instruction set architecture – RISC verses CISC
Case study : Pentium and power PC data types – operation types – addressing modes.

UNIT V**(12)****Control unit**

Control unit operations – micro operations – fetch cycle – indirect cycle – interrupt cycle – execute cycle – instruction cycle - control of the processor - functional requirements – control signals - hardwired implementation –control unit inputs and control unit logic - micro programmed control Basic concepts – Micro instructions – microprogrammed control unit – - micro instruction sequencing design considerations – sequencing techniques - address generation –micro instruction execution – micro instruction encoding

Text book:

- [1] William Stallings, “*Computer Architecture and Organization*”, PHI, Eastern Economy Seventh Edition, 2010.

Reference books:

- [1] Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “*Computer Organization*”, Fifth Edition, Tata McGraw Hill, 2011.
- [2] David A. Patterson and John L. Hennessy, “*Computer Organization and Design: The Hardware/Software Interface*”, Morgan Kaufmann, 2010.
- [3] John P. Hayes, “*Computer Architecture and Organization*”, McGraw Hill, 3rd Edition, 2012
- [4] Vincent P. Heuring and Harry F. Jordan, “*Computer Systems Design and Architecture*”, Pearson Education, 2nd Edition, 2008
- [5] M. Morris Mano, “*Computer system architecture*”, 3rd Edition, PHI.

MCA334 OOAD using UML**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

Object Oriented Analysis and Design Using UML course provides instruction and practical experience focusing on the effective use of object-oriented technologies and the judicious use of software modeling as applied to a software development process.

Learning Outcome

- To understand the object oriented life cycle.
- To know how to identify objects, relationships, services and attributes through UML.
- To understand the use-case diagrams.
- To know the Object Oriented Design process.
- To know about software quality and usability.

Unit I. (12)

Complexity – The inherent complexity of software, The Structure of complex systems, Bringing order to chaos, on designing complex systems, Categories of analysis and Design methods.

The Object Model – The evolution of object model, Elements of object model, applying the object model, Foundations of the object model.

Unit II. (13)

Classes and Objects – The nature of an object, Relationship among objects, the nature of a class, Relationship among classes, The interplay of classes and objects, On building quality classes and objects, invoking a method.

Classification – The importance of proper classification, Identifying classes and objects, Key abstraction and mechanisms, A problem of classification.

Unit III. (12)

Notation – Basic Behavioural Modelling, Basic elements, class diagram, object, state Transition diagram, Interactions, Use Case Diagrams, Activity, module and process diagrams.

Unit IV. (10)

Process – Principles, Micro and macro development process, Pragmatics- Management and planning, staffing, Release management, Reuse, Quality Assurance Metrics, Documentation, Tools, The benefits and risks and Object-oriented development.

Unit V. (13)

Architectural Modelling – Components, Deployment, Collaborations, Pattern and Frameworks, Component Diagram, Deployment Diagrams, Systems and Models.

Case Study: A domain based analysis and design using rational rose can be made.

Text Book:

- [1] Grady Booch, *Object-Oriented Analysis And Design With Applications*, Pearson Education, 3rd Edition, 2009.

Reference Books:

- [1] Mahesh P Matha, *Object Oriented Analysis and Design using UML*, PHI, 3rd reprint, 2012
- [2] Grady Booch, James Rumbaugh and Ivar Jacobson, *The Unified Modeling Languages User Guide*, Addison Wesley, 4th Edition, Reprint 2000.
- [3] Mike O'Docherty, *Object Oriented Analysis and Design Understanding system development with UML2.0*, John Wiley and Sons, 1st Edition, 2005.

MCA335 Computer Networks**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To study about network components, topologies, network models, protocols and algorithms.

Learning Outcome

Today, networks of computers are commonly used to share data and resources. The subject introduces the concept of networks, different topologies and network devices. The OSI reference model layers are discussed in detail. Error detection and correction mechanisms are dealt to give an exposure about how actually the network handles the data. The discussion about routing algorithms, congestion handling mechanisms and network security is also dealt here in this paper.

Unit I.**(12)****Introduction**

Uses of Computer Networks, Network Hardware: LAN, MAN, WAN, Wireless Network, Internetworks; Network Software: Protocol hierarchies, Design issues for the layers, Connection Oriented and Connection less Services, Service Primitives; Reference Models: OSI, TCP/IP, Comparison of OSI and TCP reference models.

The Physical Layer

Guided Transmission media: Magnetic Media, Twisted Pair, Coaxial Cable, Fiber Optics; Wireless Transmission, Brief introduction about bluetooth and wimax. Multiplexing: Frequency Division Multiplexing, Wavelength Division Multiplexing, Time Division Multiplexing; Switching: Circuit Switching, Message Switching, Packet Switching; Ethernet cabling, Manchester encoding, Differential Manchester Coding.

Unit II.**(13)****The Data Link Layer**

Data Link layer design issues, Error Detection and Correction, Elementary Data Link protocols: Unrestricted simplex protocol, Simplex stop-and-wait protocol, Simplex protocol for a noisy channel; Sliding Window protocols: One-bit sliding window protocol, Protocol using Go back N, Example Data link protocol: Higher Level Data Link Control, Data link layer in the internet;

The Medium Access Control Sublayer

The Channel Allocation problem, Multiple access protocols: ALOHA, Pure ALOHA, Slotted ALOHA, Carrier Sense Multiple Access protocols, Persistent and Non persistent CSMA, CSMA with collision detection, Collision-Free protocols: Bit map protocol, Binary countdown; Limited Contention protocols; Brief introduction to IEEE 802 standards; Ethernet MAC address, Brief introduction to Wireless LAN's, Bluetooth: Architecture, Applications, Protocol stack, Radio Layer, Bluetooth based layer, Frame structure; High-Speed LAN's, Satellite Networks.

Unit III. (12)**The Network Layer**

Network layer design issues, Routing Algorithms: Optimality principle, Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing; Congestion Control Algorithms: Congestion Prevention Policies, Jitter Control, Techniques for achieving good quality of service, Congestion control for multicasting; Internetworking, The Network layer in the Internet.

Unit IV. (11)**The Transport Layer**

The Transport service, Elements of Transport protocols: Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing, Crash recovery; A simple Transport protocol, The Internet Transport protocols: UDP, TCP.

Unit V.**The Application Layer and Network Security (12)**

Introduction to Application Layer, lossy and lossless compression techniques, Audio and Video Compression Techniques, Video on demand; Network Security: Cryptography: Introduction to cryptography, Substitution Ciphers, Transposition Ciphers, One-Time Pads, Fundamental Cryptographic Principles; Symmetric key encryption, Symmetric Key Algorithms: DES, Cipher Modes, Cryptanalysis; Public-Key Algorithms: Public-Key encryptions, RSA. Web Security: Threats, Secure Naming, Mobile Code Security;

Text Book:

- [1] Andrew S Tanenbaum ,Computer Networks, PHI publications, 5th Edition, 2012.

Reference Books:

- [1] Forouzan, Behrouz A., Mosharraf Firouz., *Computer Networks A Top-Down Approach*, TaTa McGraw Hill publications, First Edition, 2012.
[2] Stallings, William, *Data & Computer Communications*, Pearson Education Asia, 6th Edition, 2001.
[3] Prakash C. Gupta, *Data communications and Computer Networks*, 1st Edition, 5th Reprint, PHI, 2009.

MCA351 Data Structures Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

1. Implement sequential search and binary search techniques.
2. Implement Selection sort.
3. Implement Insertion sort.
4. Implement Stacks.
5. Implement Queues.
6. Implement linked lists and some operations on linked lists.
7. Write a program to convert an infix expression to the postfix form.
8. Write a program to evaluate a postfix expression.
9. Implement Quick sort.
10. Implement Merge sort for array.
11. Merge Sort a file contents (without loading the content into an internal data structure)
12. Implement Two-Way linked lists.
13. Implement Circular linked lists.
14. Implement Binary Search Tree.
15. Implement Shell sort.
16. Implement Heap sort.
17. Implement Radix sort.
18. Implement Depth First Search for Graphs.
19. Implement Breadth First Search for Graphs.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA352 UNIX Lab**Total Hours/Semester: 60****No of Hours/Week: 04****Section – A (Shell Programming)**

1. Write a shell script to print prime numbers up to a given range using arguments.
2. Write a shell script which
 - a. Converts a decimal number to binary
 - b. Converts an octal number to hexadecimal.
3. Write a shell script which merge the contents of file1, file2, file3, sort them and display the sorted output on the screen page by page.
4. Write a shell script to locate users who have logged in today or earlier but have not logged out and mail the list to root. Users who have logged more than once should appear in the list only once.
5. Write a shell script to order the file /etc/passwd on GID (primary) and UID (secondary) which would place all users with same GID together. Users with a lower UID should be placed higher in the list
6. Write a script to find the number of days between two given dates using functions.
7. Write a script to compute the factorial value with and without using recursive functions.
8. Write a shell script to search given number using binary search using function.
9. Write an awk program that reads a file and prints a report that groups employees of the same department. The following are the contents of the report
 - a. The department name in the top
 - b. All detail of the employees
 - c. Total salary for the department
10. Write an awk program which accepts input from the standard input and prints the total of any column specified as an argument.

Section – B (System Programming)

11. Demonstrate *fork()*, *kill()*, *sleep()* system calls
12. Demonstrate explicit locking and unlocking on a file using *lockf()*
13. Demonstrate process synchronization
14. Create a file and read, write operations using different child process
15. Demonstrate data sharing between process using Files
16. Implement sorting using pipes
17. Demonstrate FIFO's
18. Implement Message Queues
19. Demonstrate Semaphores
20. Demonstrate Threads

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA353 RDBMS LAB***Total Hours/Semester: 60******No of Hours/Week: 04***

1. RDBMS Lab includes an application project. The backend of the project may be any one of the following:
 - MS-SQL Server
 - Oracle
 - DB2
 - MySql
2. User interface could be made with any one of the front end tools available.
3. Students should have in-depth knowledge of the front and backend tool, which they are using.
4. Database tables are required to be normalized, at least to the second level.
5. There need to be independent forms for data entry operations.
6. All the forms in the project need to have similar look and feel in terms of background/foreground color, arrangement of controls, spacing and sizing of the controls, size of forms, etc.
7. There could be separate forms for searching purposes.
8. Master table data entry forms may include navigational buttons along with Add, Save, Delete etc.
9. Reports should be generated dynamically.

Note: Project should be developed by following software engineering process

MCA371 Seminar-I

Students have to select a topic related to the current trends and technologies in the field of Computer Science. They need to prepare the synopsis and detailed report in consultation with the faculty guide. Each Student has to give one hour presentation to their fellow classmates and to a panel of guides.

MCA431 Java Programming*Total teaching Hours/Semester: 60**No of Lecture Hours/Week: 04***Objective**

To introduce the concepts and principles of Java Programming language and to design and implement object oriented solutions to simple and complex problems. To give students experience in Java Programming and program development within an integrated development environment

Learning outcome

- An understanding of the principles and practice of object oriented programming in the construction of robust maintainable programs which satisfy the requirements.
- Competence in the use of Java Programming language in the development of small to medium sized application programs that demonstrate professionally acceptable coding and performance standards.

Unit I.**(13)****Introduction to java Programming**

History of Java. Characteristics of Java. The Java Environment – JVM, JDK & JRE. Different versions of Java. OOP Principles. Comparison of Java with C and C++.

Language Fundamentals

Data Types, Expressions, Keywords, Operators and Control Flow Statements. Structure of Java Program, Creating and Running Java Programs. Arrays.

Class and Objects

Creating class and Objects, Methods, this keyword, Constructors. Garbage Collection, the finalize() method. - Access Control. Static Blocks. Finals. Nested and Inner Classes. String Class and String Buffer Class. Command Line Arguments.

UNIT II.**(12)****Inheritance in Java**

Inheritance in classes, Using super, Method overriding, Dynamic Method Dispatch. Abstract Classes, Using final with inheritance, the Object Class.

Interfaces and Packages

Inheritance in java with Interfaces – Defining Interfaces, Implementing Interfaces, Extending Interfaces. Creating Packages, CLASSPATH variable, Access protection, Importing Packages. Interfaces in a Package.

Exception Handling in Java

try-catch-finally mechanism, throw statement, throws statement. Packages and Classes for Exception Handling

UNIT III.**(13)****Input / Output in java**

java.io package, I/O Streams, Readers and Writers, Tokenizing input, Using various I/O classes and FilenameFilter class.

Multithreading

Life cycle of a thread, Java Thread priorities, Runnable interface and Thread Class. Sharing limited Resources, Shared Object with Synchronization.

Applets

Life cycle of Applet, Applet Architecture, Applet restrictions, Creation and Execution of java Applets. Animation in Applets-Advantages of Applets. Applets Vs Applications

UNIT IV.**(11)****GUI Components (AWT & SWING)**

GUI concepts in java, Basic GUI Components in AWT, Container Classes, Layout Managers.-Flow Layout, Border Layout-Card Layout-Box Layout. Difference between AWT and SWING.

SWING

Java foundation Classes-javax.swing and Model View Controller-Creating a Frame in Swing-Displaying Image in Swing- JComponent class methods-Creating components in Swing. Writing GUI programs in java (with AWT or SWING). Event Handling- Handling Keyboard Events and Mouse Events.

SERVLETS

The life cycle of a servlet, A simple servlet, The servlet API, Servlet package, Reading servlet parameters, Handling HTTP requests and responses.

UNIT V.**(11)****Networking in Java**

Network Programming with Java.-Networking classes and Interfaces-InetAddress-Factory methods-Instance Methods. Sockets, Knowing IP address-URL-URL Connection class. Creating a server that sends data-Creating a client that receives data.-two way communication between server and client. Stages in a JDBC program-Registering the driver-Connecting to a database-Preparing SQL statements -Improving the performance of a JDBC program.

Text Book:

[1]Schildt Herbert, Java Eighth Edition: The Complete Reference, Tata McGraw-Hill, 2011

Reference Books:

[1]Deitel & Deitel, Java How to Program, Pearson Education Asia, 8th Edition, 2010

[2]Rao Nageswara ,Core Java,An Integrated Approach,Dreamtech press, 2nd Edition,2010

MCA432 Design and Analysis of Algorithms**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Pre-requisites:**

Data Structures and Algorithms

Objective

To introduce the classic algorithms in various domains.

To study the different techniques for designing efficient algorithms.

Learning outcomes

Upon successful completion of the course student will be able to

- Design efficient algorithms using the various approaches for real world problems.
- Analyze the running time of algorithms for problems in various domains.
- Apply the algorithms and design techniques to solve problems;

Unit I.**(12)****Introduction**

The role of Algorithms in Computing – Algorithms, Algorithms as a technology. Getting Started – Insertion sort, Analyzing algorithms, Designing Algorithms. Growth of Functions – Asymptotic Notations. Recurrences – The Substitution method, Recursion Tree method and Master method.

Unit II.**(13)****Divide and Conquer**

General Method, Binary Search, Finding the Maximum and Minimum, Merge Sort, Quick Sort, Selection sort, Strassen's Matrix Multiplication.

Greedy Method

Knap Sack Problem, Minimum Spanning Trees, Prim's algorithm and Kruskal's algorithm.

Unit III**(13)****Dynamic programming Method**

Optimal Binary Search Trees, Traveling Salesman Problem, Longest Common Subsequence Back Tracking

Introduction - The 8-queens problem, Sum of Subsets

Branch n Bound

General Method- Traveling Salesman Problem

Unit IV.**(11)****Graph Algorithms**

Representation of Graph, Depth First Search, Breadth first search.

Single Source shortest path – Dijkstra's Algorithm and Bellman Ford Algorithm. All Pair Shortest Path – Floyd-Warshall Algorithm.

Lower Bound Theory

Comparison trees for sorting and searching.

Unit V.

(11)

NP-Hard and NP-Complete problems

Basic Concepts, NP_Hard graph problems, NP-Hard Scheduling problems, NP-Hard code generation problems, some simplified NP-Hard problems

Text Books:

- [1] Cormen T H, Leiserson C E, Rivest R L and Stein, Clifford, *Introduction to algorithms*, PHI, 2nd Edition, 2009.
- [2] Horowitz E and Sahni S. *Fundamentals of Computer Algorithms*, Computer Science Press, 2008.

Reference Books :

- [1] Gelder Van Allen and Baase Sara, *Computer Algorithms – Introduction to Design and Analysis*, Addison Wesley, 3rd Edition, 2002.
- [2] Aho A V, Hopcroft J E and Ullman J D., *The Design and Analysis of Computer Algorithms*, Addison Wesley Publishing House, 1983.
- [3] Dromey, R.G., *How to solve it by Computer*, Prentice-Hall International, 2006.

MCA433 Mobile Applications**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

This course aimed at helping learners create applications using Google's Android™ open-source platform. The course explains what Android™ is and how it compares to other mobile environments, the setup of the Android™ Eclipse-based development tools, the Android™ SDK, all essential features, as well as the advanced capabilities and APIs such as background services, accelerometers, graphics, and GPS.

Learning outcome

Upon completion of this course, students will be able to:

- Build your own Android apps.
- Explain the differences between Android and other mobile development environments.
- Design and develop useful Android applications with compelling user interfaces by using extending and creating your own layouts and views and using menus.
- Secure, tune, package and deploy Android Applications.

Unit I.**(12)****Introduction**

Brief History of Embedded Device Programming , Introduction to Android , Get to know the required tools , Creating your first Android application , Anatomy of android Application. Understanding Activities, linking Activities using intents, fragments, calling Built-in Applications using Intents, Displaying Notifications.

Unit II.**(12)****User Interface and Designing with views**

Understanding the components of a screen, adapting to display orientation, managing changes to screen orientation, Utilizing the Action Bar, Creating the user Interface programmatically, Listening for UI Notifications.

Using Basic Views, Using Picker views, Using List views to display lists, Understanding specialized fragments.

Unit III.**(12)****Displaying with views , Data persistence and Content Providers**

Using Image Views to display pictures, using menus with views, some additional views. Saving and loading user preferences , persisting Data Files, Creating and using Databases. Sharing Data in Android, using content provider, creating your own content providers, using content providers.

Unit IV.**(12)****Messaging , Location based services and Networking –**

SMS Messaging , Sending E-mail, Displaying Maps, Getting Location Data, Monitoring a

Location. Hands on project : Building a Location Tracker. Consuming Web Services using HTTP, consuming JSON Services, Basic Socket Programming

Unit V.**(12)**

Creating your own services , Establishing Communications between a service and an activity, binding activities to services, understanding Threads. Preparing for Publishing, Deploying APK Files.

Text Book:

- [1] Wei-Meng Lee, “*Beginning android 4 application Development*, John Wiley & sons, Inc, 2012

Reference:

- [1] Paul Deitel-Harvey Deitel-Abbey Deitel-Michael Morgano,*Android for Programmers An App-Driven Approach*”,Pearson Education Inc., 2012.
[2] Jerome (J.F) DiMarzio , "Android - A programmer's Guide", TataMcgraw Hill ,2010, ISBN: 9780071070591.

MCA441A Digital Image Processing**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

The Objective of this course is to cover the basic theory and algorithms that are widely used in Digital image processing. Develop hands-on experience in using computers to process images with Matlab image processing toolbox.

Learning outcome

Upon successful completion of the course the student would

- Understand the theoretical background of Image processing.
- Apply image enhancement, restoration, compression and segmentation in both frequency and spatial domain.
- Represent and recognize objects through patterns in application.

Unit I. (12)**Introduction and Digital Image Fundamentals**

The origins of Digital Image Processing, Fundamental Steps in Image Processing, Elements of Digital Image Processing System, Image Sampling and Quantization, Basic relationships: Neighbors, Connectivity, Distance Measures between pixels, Linear and Non Linear Operations.

Unit II. (12)**Image Enhancement in Spatial Domain**

Gray Level Transformations, Histogram Processing, Histogram equalization, Histogram specification, Basics of Spatial Filters, Smoothing and Sharpening Spatial Filters.

Image Enhancement in Frequency Domain

Introduction to Fourier Transform and the frequency Domain, Smoothing and Sharpening, Frequency Domain Filters, Homomorphic Filtering.

Unit III. (12)**Image Restoration and Image Compression**

A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise, Periodic Noise Reduction by Frequency Domain Filtering. Image Compression models: Huffman coding, Run length coding, LZW coding.

Unit IV. (12)**Image Segmentation and Representation**

Point, Line and Edge detection. Thresholding – Basic global thresholding, optimum global thresholding using Otsu's Method. Region Based Segmentation – Region Growing and Region Splitting and Merging. Representation – Chain codes, Polygonal approximations using minimum perimeter polygons.

Unit V.**(12)****Description and Object Recognition**

Boundary descriptors – Fourier descriptors. Regional descriptors – Topological descriptors and Moment invariants. Introduction to Patterns and Pattern Classes. Decision-Theoretic Methods – Minimum distance classifier, K-NN classifier and Bayes' classifier.

Text Book:

- [1] R. C. Gonzalez & R. E. Woods, Digital Image Processing, 3rd Edition. Pearson Education, 2009.
- [2] A.K. Jain, Fundamental of Digital Image Processing, 4th Edition. PHI, 2011.
- [3] Rafael C. Gonzalez, Richard E. Woods and Steven L Eddins, Digital Image Processing Using MATLAB, 2nd Edition. PHI, 2009.

Reference Books:

- [1] M. A. Joshi, Digital Image Processing: An algorithmic approach, 2nd Edition. PHI 2009.
- [2] B.Chanda, D. DuttaMajumdar, Digital Image Processing and analysis, 1st Edition, PHI, 2011.

MCA441B Advanced Database Management System**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To provide strong foundation for database application development, appreciate the underlying core database engine and emerging database technology.

Learning Outcome

- Ability to create Stored Database Procedures for writing consistent, well tuned backend code
- Appreciate Query Optimisation in the Database Engine
- Understand the need for Document Oriented Database for Distributed System
- Able to consolidate theoretical database understanding
- Insights into recent developments in database technologies

Unit I. (12)

PL/SQL Fundamentals: Variables, Reserve Words, Identifiers, Anchored Data types, Block, Labels. Use of DML in PL/SQL, Commit, Rollback, Savepoint.

Conditional Control: IF, CASE, NULLIF, COALESCE

Iterative Processing with Loops: Loop Basics, Simple Loop, While, For (Numeric and Cursor)

Unit II. (12)

Exception: Scope, User Defined Exception, Exception Propagation, Raise Application Error, Exception_Init, SQLCODE, SQLERRM

Data Retrieval: Cursor, Use of Record Types, Explicit Cursors, Cursor Attributes, Cursor For Loop, for update and where current cursors

Procedures: Creating Procedures, Query Data Dictionary for Information on Procedure, IN and OUT parameters, Creating and Using Functions

Unit III. (13)**The Relational Algebra and Relational Calculus**

Unary Relational Operations SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relation Operators: JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra

Query Optimization

Translating Queries into relational algebra, Algorithms for external sorting, Algorithms for Select, Join, Project, Implementing aggregate operation and outer join, Combining operation, Using heuristic in query optimization, Using selectivity and cost, estimates in query optimization, Query Optimization in oracle

Unit IV. (10)**Enhanced Data Models**

Active database concepts and triggers, Temporal database concepts, Spatial and Multi Database, Introduction to Deductive Database

Emerging Database Tech and Applications

Mobile Database, Multimedia Database, Geographic Information System, Genome Data Management

Unit V. (13)

Case Study for a Document Oriented Database as a standalone or for high volume distributed applications. “Couchdb”

Introduction, Why CouchDB, Eventual Consistency: Grain, CAP Theorem, Local Consistency, Distributed Consistency. Core API: Server, Database, Documents, Replications. Design Documents, Finding Data with Views, Validation Functions, Show Functions

Text Book:

[1] Benjamin Rosenzweig & Elena Silvestrova Rakhimov, *Oracle PL/SQL by Example*, Pearson, 4th Edition, 2009. (for Unit I and II).

[2] Elmasri & Navathe, *Fundamentals of Database Systems*, Addison-Wesley, 6th Edition, 2010. (Unit III & IV).

[3] Anderson, Lehnardt & Slater, *Couchdb, The Definitive Guide*, O'Reilly, 1st Edition, 2010 (Unit V).

Reference Book

[1] Steven Feuerstein, *Oracle PL/SQL Programming*, O'Reilly, 5th Edition, 2010

MCA441C Multimedia System and Applications

Total teaching Hours/Semester: 60

No of Lecture Hours/Week: 04

Objectives:

- This course will provide the students with an overview of multimedia technologies and the latest developments in multimedia systems.
- Students will be able to gain valuable hands on experience in multimedia systems and applications.
- Issues in effectively representing, processing, and retrieving multimedia data will also be addressed.
- Recent multimedia papers or technique reports will be presented or assigned as homework.

Learning outcome:

- Comprehend multimedia system fundamentals.
- Design and implement a multimedia application or identify a problem in certain multimedia area and provide a reasonable solution.

Unit I. (12)

Multimedia Information representation

Introduction, Definition of Multimedia, Digitization principles- Analog signals, Encoder design, Decoder design, Text-Unformatted text, Formatted text, Hypertext, Images- Graphics, digitized documents, digitized pictures, Audio – PCM speech , CD quality audio, Synthesized audio. Video- Broadcast television, digital video, PC video, Video Content.

Unit II. (13)

Text and Image Compression

Introduction, compression principles- Source encoders and destination decoders, Lossless and lossy compression, entropy encoding, source encoding. Text compression- static and dynamic Huffman coding, Arithmetic coding, Image compression-Graphics interchange format, Tagged image file format, digitized documents, digitized pictures, JPEG.

Unit III. (13)

Audio and Video Compression

Introduction, Audio compression, Frequency, amplitude, sample rate, Differential pulse code modulation, Adaptive differential PCM, Adaptive predictive coding, Linear predictive coding, code-excited LPC, perceptual coding, MPEG-MP3 audio coders, Dolby audio coders.

Video compression principles, video Standards: NTSC, PAL, SECAM, Inter-frame, Intra-frame, video encoding, algorithms H.261, H.263, MPEG, MPEG1, MPEG2, MPEG4, Video for WEB

Unit IV. (12)

Standards for Multimedia communications

Reference models-TCP/IP, Protocol basics, standards relating to interpersonal communications, Circuit mode networks, Packet-switched networks, Electronic mail, standards

relating to interactive applications over the Internet, information browsing, Electronic commerce, intermediate systems, Java and Java Script, Standards for entertainment applications, Movie/Video on demand, Interactive television.

Unit V.**(10)****Multimedia Applications**

Understanding Designing and implementations of interactive applications, entertainment applications, Multimedia in internet and Web, Video Emails, video conferencing, Web casting, Software for image editing and Compression, Audio editing and compression, Video editing and compression, Voice recognition applications, Gesture based applications, interactive games design and implementation.

Text Books

- [1] Fred Halshall, *Multimedia communication-application, network, protocol and standards*, 1st Edition, Pearson Education Ltd, 2009
- [2] Ralf Steinmetz, Klara Nahrstedt, *Media Coding and Content Processing*, Volume I, PHI, 2011

Reference Books

- [1] Krishna Kumar, *Multimedia communication*, Dorling Kindersley Publishers, Pearson Education, 2008
- [2] Nigel Chapman and Jenny Chapman, *Digital Multimedia*, John Wiley & Sons Ltd., 2009.

MCA441D Software Quality and Testing**Total teaching Hours/Semester: 60****No of Lecture Hours/Week:60****Objective**

To understand the need for Software Quality, Tools Used and Metrics involved. To appreciate software testing principles and methods to detect in the ever changing software technological changes

Learning Outcomes

- Fundamental concepts of Software Quality and Testing
- Ability to test code, artifacts better
- Learn to apply different Quality Tools
- Understand the advantages of Extreme Testing and High Order Testing
- Create effective test plan
- Create detailed test cases
- Appreciate the need for Software Quality Metrics and Assessments

Unit I. (12)**Introduction to Software Quality, Framework and Quality Standards**

Quality: popular view, Quality: professional view, software quality, total quality management, The defect prevention process, process maturity framework and quality standards (CMM , SPR Assessment, Malcolm Bridge, ISO9000)

Unit II. (12)**Fundamentals in Measurement Theory**

Levels of measurement some basic measures, reliability and validity

Software quality metrics

Product Quality Metrics, in-process quality process, example of Metrics Program –Motorola, HP

Unit III. (12)**Seven Basic Quality Tools**

Ishikawas' seven basic tools, checklist, pareto diagram, histogram, runchart, scatter diagram control chart cause and effect diagram.

Defect Removal Effectiveness

Literature review, a close look at DRE, defect removal effectiveness and quality planning

Unit IV. (12)**Fundamentals of Software Testing**

Software Testing Principles, Economics of Testing

Inspection and walkthrough, code inspection, an error checklist for Inspection, Walkthroughs, Desk Checking, Peer Rating

Module Testing

Unit V.

(12)

High Order Testing, Debugging and Extreme Testing

High Order Testing - Debugging by Brute Force, Induction, Deduction, Backtracking
Extreme Programming basics, Extreme Testing, Extreme Testing Applied

Text Book:

- [1] Stephen H Kan, *Metrics and Models in Software Quality Engineering*, 2nd Edition, reprint 2006
- [2] Glenford J. Myers, *The Art of Software Testing*, John Wiley and Sons publications, 2004

Reference books:

- [1] S A Kelkar, *Software Quality and Testing*, PHI, 1st Edition, 2012.

MCA441E Microcontroller and Applications**Pre-requisite**

Basic Knowledge of Microprocessors and Digital Electronics

Learning Objective

The course objective is to provide sufficient detailed knowledge of a microcontroller and its applications.

Outcome

At the conclusion of course, students are expected to be able to:

- Understand and use various IO devices such as: keypads, A to D converters,
- Understand details on basic I/O drivers and microcontroller device interfaces, port programming,
- Understand the basic types of memory used in microcontrollers;
- Understand the hardware and software resources required for real-time
- Microcontroller applications

Unit I (12)

Introduction: Introduction to Micro controllers and Embedded processors, Overview of the 8051 family

8051 Assembly Language Programming:

Introduction to 8051 Assembly Programming, Data types and Directives, 8051 Flag bits and PSW register , 8051 Register Bank & stack.

Unit II (12)**Instructions and programming**

Address modes, JUMP, CALL and LOOP instructions, Arithmetic Instructions and Programming, Logical Instruction and Programming, Single bit instruction & Programming

I/O Port Programming:

Pin description of 8051, I/O programming: Bit manipulation

8051 Addressing modes:

Immediate and Register Addressing modes, Accessing memory using various addressing modes

Unit III (12)**Arithmetic Instructions and programs:**

Unsigned addition and subtraction, unsigned multiplication and division, signed number concepts and arithmetic operations

Logic Instructions and programs:

Logic and compare instruction, Rotate and Swap Instructions, BCD and ASCII application program

Unit IV (12)**Single bit Instructions and Programming:**

Single bit Instruction programming, Single bit Instruction operation with CY, reading input Pins vs Port Latch

Timer/Counter Programming in the 8051

Programming 8051 Timers, Counter Programming.

Unit V**(12)****8051 Serial Communication**

Basics of serial communication, 8051 connection to RS232, 8051 serial communication programming

Real world Interfacing: LCD, ADC and sensors, stepper motor, keyboard, DAC, Interfacing to external memory;

Text Books:

1. Muhammed Ali Mazidi & Jannice Gillespie Mazidi, *The 8051 Microcontroller and Embedded systems*, Pearson Education, 2nd Edition, 2007.
2. Embedded Micro Computer System - Jonathan W. Valvano

Reference Books:

3. Mike Predko, *Programming and Customizing 8051*, Tata McGrawHill, 1st Edition, 2000
4. Stephen Brown et al., *Fundamentals of Digital logic and very low design*, Pearson Education, 4th International Edition, 2008 .

MCA442A WEB ENGINEERING**Total teaching Hours/Semester: 60****No of Lecture Hours/Week:60****Course objective**

The World Wide Web has become a major delivery platform for information resources. Many applications continue to be developed in an ad-hoc way, contributing to problems of usability, maintainability, quality and reliability. This course examines systematic, disciplined and quantifiable approaches to developing of high-quality, reliable and usable web applications. The course introduces the methodologies, techniques and tools that support their design, development, evolution, and evaluation.

Learning outcome

Upon successful completion of the course the student will:

- Be able to understand the concepts, principles and methods of Web engineering.
- Be able to apply the concepts, principles, and methods of Web engineering to Web applications development.
- Be familiar with current Web technologies.
- Be familiar with Web application development software tools and environments currently available on the market.
- Be able to understand the technologies, business models and societal issues of Web 2.0/Web 3.0 and Semantic Web.

Unit I.**(12)****History of the Web**

The arrival of the Browser, flattening of the world, from linking to searching, commercialization of the Web, Peer-to-Peer networks and free file sharing. Technological development- IP networking, HTML and XML, Web services and RSS.

Socialization of the Web -Blogs and wikis, Social networks. Arrival of Web 2.0.

Developments in Web Technology - HTML,XML,CSS, Scripting technologies, Web applications of XML-Web services, Web feed forms, P2P file sharing networks, other P2P applications.

Unit II.**(12)****Enabling Techniques and technologies**

Rich Internet Applications-sample RIAs email applications, XMLHttpRequest, the link needed for Ajax, More RIA- Office and Map applications. API.

WPCs and their mach-ups

The situation in Web 1.0, content syndication with Web feeds, Mach-ups based on WPCs, Tagging- Flickr and tagging, Social bookmarking, Folksonomies.

Unit III. (12)**Framework for Application development**

Development methodologies, client-side Ajax toolkit, development of Zimlet using AjaxTK, Server-side frameworks-Ruby on Rails, Creating web applications with rails, Framework for RIA technologies-RIA development with OpenLaszlo, Flash versus Ajax.

Impacts of Next Generation of the Web- Commission-based brokerage and merchants, Advertising, Information intermediaries, The Community and the subscription models, Data Ownership.

Software as a Service(SaaS) - A look back ASP model, Provider oriented View, The Consumer-oriented view and service customization.

Unit IV. (12)**Socialization or co-creation of content**

Social search, social aspects of software, Impacts of online social networks, user generated content in advertising. Second life.

Semantic Web and Web 2.0/Web 3.0

Search revisited, Data and information integration, The semantic Web idea, the structure of the Semantic Web, Languages of the semantic Web- The Resources Description Framework (RDF), RDF Schema (RDFS), Web ontology language(OWL).

Ontologies – Introduction, Design of an ontology, OntoMedia: an ontology-based personal entertainment system.

Unit V. (12)**Web Security**

Security issues , causes, Solutions, Design Patterns and best practices

Case Study

Web – Smartphone Applications using native coding and Cross-platform technology using HTML 5, CSS3 and JavaScript.

Learning about Windows 8 Apps, Android Apps, Google Chrome Web Store Apps.

Hands on Experience with Phone Gap, Sencha and Titanium platforms.

Text Books:

- [1] Gottfried Vossen, Stephan Hagemann, *Uleashing Web2.0-from concept to Creativity*, Elsevier , 1st Indian Reprint, 2009
- [2] Greti Kappel, Birgit Proll, Sieghried Reich, Werner Retschitzegger, *Web Engineering* , John Wiley & sons, Ltd, 1st Edition, 2010 reprint.

Reference Books:

- [1] Roger S Pressman, David Lowe, *Web Engineering* , Tata McGrawHill, 1st Edition, 2008 reprint.
- [2] Neil Daswani, Christoph Kern, Anita Kesavan, *Foundations of Web Security*, APRESS,2007, ISBN:1-590590784-2.

MCA442B Network Security**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 60****Objective**

To make the students learn the principles and practices of Cryptography, Network Security and to enable the students understand the various methods of encryption and authentication and help them identify the application of these techniques for providing Network and System Security.

Learning Outcome

At the completion of the course, the students should be able to

- Understand the principles and practices of Cryptography and Network Security.
- Describe the five keys of Network Security.
- Appreciate the role played by Cryptographic techniques in enhancing Network and system Security.
- Identify and explain the concepts, protocols and technologies associated with a secure communication across the Network and the Internet.
- Discuss the objectives of authentication and access control methods and describe how the available methods are implemented in the defense of a network.

Unit I.**(13)****Concepts of Security & Classical Encryption Techniques**

Introduction, The need for security, Security Approaches, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security. Symmetric Cipher Models – Substitution techniques, Transposition techniques, Steganography, Block Cipher Operation, Electronic Code Book, Cipher Block Chaining, Block Cipher Principles, The Data Encryption Standard, A DES Example, The Strength of DES, Evaluation criteria for AES, AES Cipher.

Unit II.**(12)****Public Key Cryptography and Cryptographic Hash Functions**

Introduction To Number Theory, Modular Arithmetic, Prime Numbers, Euler's Totient Function, Principles of Public Key Cryptosystems, The RSA Algorithm, Other Public key cryptosystems, Diffie Hellman Key Exchange.

Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Hash Functions Based on Cipher Block Chaining, MD5 Message Digest Algorithm, Secure Hash Algorithm SHA 512.

Unit III.**(11)****Message Authentication Codes and Digital Signatures**

Message Authentication Requirements – Message Authentication Functions –Requirements for Security of MACs, MACs Based on Hash Functions, HMAC, MACs Based on Block Ciphers, Data Authentication Algorithm.

Digital Signatures, Elgamal Digital Signature Scheme, Schnorr Digital Signature Scheme, Digital Signature Standard.

Unit IV.**(12)****Key Management & Distribution And User Authentication**

Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public Key Infrastructure.

Remote user Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Kerberos, Motivation, Kerberos Version 4, Remote User-Authentication Using Asymmetric Encryption, Federated Identity Management.

Unit V.**(12)****Network & Internet Security**

Transport-Level Security – Web security Considerations, Secure Socket Layer and Transport layer Security; **E-Mail Security** – Pretty Good Privacy, S/MIME.

IP Security – IP Security Overview, IP Security Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange. **E-Commerce Security** – SET Overview, Key Features of SET, SET Participants, Dual Signature.

Text Books:

[1] William Stallings, *Cryptography and Network Security*, Prentice Hall, 5th Edition, 2010.

Reference Books:

[1] Atul Kahate, *Cryptography and Network Security*, Tata McGraw-Hills, 8th Reprint, 2006.

[2] Brijendra Singh, *Network Security and Management*, PHI, 3rd Edition, 2013

[3] Eric Maiwald, *Information Security Series, Fundamental of Network security*, Dreamtech press
2004.

[4] Charlie Kaufman, Radia Perlman, Mike Speciner, *Network Security: Private communication in public world*, Prentice Hall, 2002.

MCA442C Data Warehousing**Total teaching Hours/Semester: 60****No of Lecture Hours/Week:60****Objective**

To understand the concepts of Data Warehouse system.

Learning Outcome

On successful completion of this course, the students should be able to:

- Discuss the role of data warehousing and enterprise intelligence in industry and government.
- Summaries the dominant data warehousing architectures and their support for quality attributes.

Unit I.**(12)**

Introduction to Data Warehouse: Basic elements of the Data Warehouse: Source system-Data staging Area-Presentation Server-Dimensional Model-Business process-Data Mart-Data warehouse.

Data Warehouse Design: The case for dimensional modeling – Putting Dimensional modeling together: the data warehouse bus architecture – Basic dimensional modeling techniques.

Unit II.**(12)**

Data Warehouse Architecture: The value of architecture – An architectural framework and approach – Technical architecture overview – Back room data stores – Back room services. Back Room Services.

Data Staging: Data staging overview – Plan effectively – Dimension Table staging – Fact Table loads and warehouse operations – Data quality and cleansing – issues.

Unit III**(12)**

Metadata: Metadata, metadata interchange initiative, metadata repository, metadata management, implementation examples, metadata trends, reporting and query tools and applications- tool categories, the need for applications.

OLAP: Operational Data Store-OLAP: ROLAP, MOLAP and HOLAP. Need for OLAP, multidimensional data model, OLAP guidelines, multidimensional versus multi relational OLAP, categorization of OLAP tools.

Unit IV.**(12)**

Building a data warehouse: Business considerations, Design considerations, technical considerations, implementation considerations, integrated solutions, benefits of data warehousing, Relational data base technology for data warehouse, database architectures for parallel processing, parallel RDBMS features, alternative technologies

Unit V.**DBMS schemas for decision support****(12)**

Data layout for best access, multidimensional data model, star schema, STARjoin and STARindex, bitmapped indexing, column local storage, complex data types, Data extraction, clean up and transformation tools-tool requirements, vendor approaches, access to legacy data, vendor solutions, transformation engines

Text Book:

[1] Kimball Ralph, Reeves, Ross, Thronthwaite, „*The Data warehouse lifecycle toolkit*”, Wiley India, 2nd Edition, 2006.

[2] Berson Alex, Stephen J Smith, “*Data Warehousing, Data Mining and OLAP*”, TATA McGraw-Hill, 13th reprint 2008.

Reference book:

[1] Soumendra Mohanty, „*Data Warehousing design, development and Best practices*”, TATA McGraw-Hill, 4th reprint 2007.

MCA442D Linux Administration**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

This course provides a practical introduction to Linux system Administration. It helps students gain knowledge and skills required for the role of Linux system administrator.

Learning outcome

Upon successful completion of the course students would be able to

- Understand the role and responsibilities of a Linux system administrator
- Install and configure the Linux operating system
- Manage the resources and security of a computer running Linux at a basic level
- Configure and manage simple TCP/IP network services on a Linux system

Unit I.**(12)****Installation and Configuration**

Duties of System Administrator, Standard Installation: Exploring Components, Checking supported Hardware, Creating the Boot Disk, Starting the Installation, Partitioning the Hard Disk, Using Disk Druid, Configuring the Installation, Package Installation. System Startup and Shutdown: Examining the boot process, Exploring Run-levels, Starting Programs at System Boot, Shutting down the System, GRUB Configuration. The File System Explained: Understanding File System Structure, Working with Linux File System, Memory and Virtual File System, Linux Disk Management.

Unit II.**(11)****Expanding the System**

Installing and Upgrading Software Packages: Using Package Manager, Checking versions, Obtaining newer softwares, Installing software from source. Devices and Modules: Hardware Device Installation, Device Information, udev Device Files, Hardware Abstraction Layer, Manual Devices, Installing and Managing Terminals and Modems, Input Devices, Installing Sound, Network and Other cards, Modules- Kernel Module Tools, Managing Modules, depmod, modprobe, insmod and rmmod commands, Installing new modules.

Unit III.**(12)****Linux Management**

Configuring System at the Command Line: Managing Processes, Maintaining the File System, Time Keeping, Automating Scripts using at and cron jobs. Administering Users and Groups: Administering User Accounts, Working with Group Accounts, Understanding the Root Account, Implementing Sudo, Using File System Quotas

Unit IV.**(13)****Linux Management(Cont.)**

Backing Up and Restoring the File System: Creating a Backup Plan, Choosing Media for backup Understanding backup Methods, Using Backup Tools – Command line tools and

AMANDA tools. Performance Monitoring: Tools, Measuring Memory Usage, Viewing Running Tasks using ps and top, Monitoring I/O Activity, Using sar.

Networking

Managing the X Window System: Configuring X Server – Setting Display Resolution and Changing Video Card Type. TCP/IP Networking: TCP/IP explained, Understanding and Setting up Network Interface Card (NIC), Working with Gateways and Routers, Configuring DHCP Server and Client, Editing Network Configuration.

Unit V.**(13)****Networking(Cont.)**

Network File System: NFS Overview, NFS, Installation, Configuring NFS Server, Configuring NFS Client, Using Automount Services. Network Information System: Understanding , Planning and Configuring NIS Server and NIS Client. Installing Samba, Creating Samba Users, Starting Samba Server and Connecting to Samba Client. Configuring BIND: DNS- Understanding DNS, Configuring server files, Checking the configuration. Configuring Sendmail. Configuring FTP Services.

Text Book:

[1] Collings Terry and Wall Kurt, *Red Hat Linux Networking & System Administration*, Wiley Indian, 3rd Edition, reprint 2009.

[2] Petersen Richard, *The Complete Reference: Fedora 7 & Red Hat Enterprise Linux*, Tata McGraw Hill Edition, 2007.

Reference Books:

[1] Richard Peterson, *Redhat 6 :Desktop and Administration* , Surfing Turtle press , 2010.

[2] Soyinka Wale, *Linux Administraiton: A Beginner's Guide*, 5th Edition, 2008.

MCA442E ADVANCED MICROPROCESSOR**Objective**

A course emphasizing applications of microcomputers to dedicated hardware functions and introduce the basic concepts of microprocessor and assembly language programming.

Learning Outcome

At the conclusion of this course, student will be able to demonstrate:

- Knowledge and skill in the application of parallel ports, counter/timers, programmable counter arrays, serial ports, and interrupts.
- The objective of this course is to provide extensive knowledge of microprocessor based systems and interfacing techniques.

Unit I.**Introduction to Pipelined Processors (12)**

Pipelining: An Overlapped Parallelism, Linear pipelining, Classification of Pipelined Processors, Principles of designing pipeline processor, Data computers, Systolic architecture, Superscalar, Super pipeline and VLIW processors.

Unit II.**(12)****Intel 80386DX Processor:**

Detailed study of Block diagram, Signal interfaces, Bus cycles, Programming model, Operating modes, Address translation mechanism in protected mode, Memory management, Protection mechanism..

Unit III.**(12)****Intel P5 Micro architecture**

Pentium Processor Block diagram, Superscalar operation, Integer pipeline stages, Floating point pipeline stages, Branch prediction logic, Cache unit.

Intel P6 Micro architectures

Introduction to Pentium-Pro Processor, Special Pentium-Pro features, Introduction to Pentium-2 Processor, Pentium-2 software changes, Pentium-3 processors.

Unit IV.**(12)****Pentium-4 & IA-64 Architectures**

Pentium-4 Net Burst Architecture, IA-64 Itanium Processor architecture

Sun SPARC Architecture

SPARC Processor, Data Formats, Registers, Memory model. Study of SuperSPARC and UltraSPARC architectures

Unit V.**(12)****Study of System Buses**

Features, classifications, applications of the system buses like ISA, ATA, SCSI, PCI and USB. (Study of the buses is without signals and the timing diagrams),.

Case study on Different existing and new Architectures.

Text Books:

- [1] Kai Hwang, Faye A, Briggs, Computer Architecture and Parallel Processing, McGraw Hill Education, 1st Edition, 2012
- [2] Don Anderson, Pentium Processor system Architecture, Addison-Wesley Professional, 2nd Edition.
- [3] Barry B Brey, The Intel Microprocessors: Architecture, Programming and Interfacing, Pearson, 2008.

Reference Books:

- [1] K M Bhurchandi, ,Advanced Microprocessor and Peripheral, McGraw Hill Education, 2012.

MCA451 Java Programming Lab***Total Hours/Semester: 60******No of Hours/Week: 04*****Section – A**

1. Write a program to demonstrate various data types and operators.
2. Write a program to implement command line arguments
3. Demonstrate method overloading and constructor overloading.
4. Demonstrate the usage of static keyword in java – use static data and static block.
5. Demonstrate final keyword with respect to variable, method and class.
6. Write a program to demonstrate the usage of the keywords this & super.
7. Demonstrate abstract class.
8. Demonstrate the usage of interface for multiple inheritance.
9. Differentiate the usage of throw, throws and try-catch-finally by writing a java program.

Section – B

10. Demonstrate various I/O streams in java.
11. Demonstrate the Reader/Writer classes in java.
12. Demonstrate the multithreading concept by implementing Runnable interface.
13. Demonstrate the multithreading concept by extending Thread class.
14. Write an applet program and using paint function make some graphics.
15. Write a program to demonstrate the usage of different Layouts in java.
16. Write a java program to demonstrate various GUI components in java (AWT / SWING) with appropriate Event Handling.
17. Write a program to implement two way communication between server and client.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA452 Computer Architecture Project Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

Students should be divided into batches, each batch containing not more than 3 students. It can be either hardware-software or simulation software related to CA principles/algorithms. This lab is intended to practice the principles (theory) that they have studied in Computer architecture. It is entertained to do a hardware related project incorporating innovative/new ideas.

Some of the project titles:

Processor:

- Design of Micro control unit
- Design of LCM processor
- Design of Arithmetic unit
- Design of logic unit
- Design of shift unit
- Design of nano control unit
- Design processor for Recursive functions
- Design of Floating Point Operation processor
- Design of disassembler (converting from object code to Assembly language code)

Memory:

- Memory interleaving
- Design of Associative cache
- Design of Direct mapping
- Design of Set Associative cache
- Virtual memory
- Look-aside cache
- Look-through
- Cache snooping

Bus arbitration:

- Daisy chaining
- Polling
- Independent Request
- DMA control

Pipeline:

- Instruction pipeline
- Arithmetic pipeline
- Vector processing

Fault tolerance system

MCA453A DIGITAL IMAGE PROCESSING Lab

Total Hours/Semester: 60

No of Hours/Week: 04

1. Reading, Writing and Displaying images
2. Generating and Plotting Image Histograms
3. Histogram Equalization
4. Linear spatial filters (average and Laplacian)
5. Removal of image noise through non-linear filters (median filters)
6. Displaying R, G and B components from a colour image
7. Run length encoding and decoding algorithm
8. Line and Edge detection (Sobel, Prewitt and Roberts)
9. Global thresholding algorithm to segment gray scale images
10. Pattern Matching using Minimum Distance Classifiers

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA453B ADBMS LAB**Total Hours/Semester: 60****No of Hours/Week: 04****Objective**

To provide hands on experience in Oracle SQL and PL/SQL, enhance previous RDBMS project with a stored procedure and partial implementation using Document Oriented Database

Learning Outcome

- Think Logical to solve SQL Questions
- Appreciate performance of different types of queries
- Learn to create Stored Database Procedures for writing consistent, well tuned backend code
- Understand the need for Document Oriented Database for Distributed System
- Able to consolidate theoretical database understanding

1) Select Queries

- a. Retrieving Data from Multiple Tables using different joins
- b. Group By and Having
- c. Sub-Queries and Correlated Subqueries

2) Data Manipulation Language (DML)

- a. Insert, Update, Delete
- b. Locking tables

3) PL/SQL

- a. Variables and type declarations
- b. Loop structure
- c. PL/SQL language commands
- d. Cursor/ Cursor loops
- e. Exceptions

4) Create a stored procedure which needs to select multiple tables and insert and update some tables for the RDBMS project done in previous semesters. Should use Locking, Commit, Rollback and Savepoint. Examples could be Interest Calculation, Closing of Fixed Deposit, Account Opening etc**5) Couchedb**

- a. Install Couchedb
- b. Implement Database Design using Couchedb for a part of previous semester RDBMS project
- c. Implement backend and populate data for a Module
- d. Modify some forms to handle couchdb as the backend

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA453C Multimedia Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

Photoshop – Image Editing, Graphic designing and Image Compression,
Sound forge/Sound booth FL Studio - Audio editing and Audio Compression
Premiere / After effects - Video editing and Special effects
Flash/Flash Media server - Interactive Presentation and Application, Games

1. Design a Brochure for a given product and details, learn about different Image file formats.
2. Design a poster with given information and learn about Image compression.
3. Learn to prepare images for Print, Web and Video.
4. Edit the sound file and Learn about Effects and Filters of sound.
5. Record Your voice and learn about Audio Compression.
6. Record an Audio Program and Learn about streaming an audio content.
7. Learn about Video editing – Prepare video with rough cut.
8. Prepare the video for different Standards (NTSC/PAL/SECAM....etc).
9. Prepare video content with title and special effects.
10. Record video content and learn about video compressions.
11. Prepare Video content for streaming.
12. Prepare an interactive presentation using flash.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA453D Software Quality and Testing Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

1. Implement checklist for Design Review for RDBMS Projects.
2. Review few project to check for Non Compliances for currently developed projects by the students in the UG Courses in the department. Based on the checklist created for Design review
3. Implement the following Quality Tools for a hypothetical project
 - Pareto Diagram
 - Histogram
 - Runchart
 - Scatter Diagram
 - Control Chart
4. Create a root cause analysis for a current problem (eg Why India is not doing well in Hockey?, Why do students not do well in exams ?)
5. Create a Test Plan for release an Mobile Android OS in the market
6. Implement 50 test cases for one project done by the student in the previous semester
7. Do a code review and walkthrough of 5 Data Structure Program in previous semester by students of the same class
8. Create Auto Test Cases Scripts to test code in C or Java for
 - “Binary Search” in an Array
 - “Find the second largest number in three numbers”
9. Write JUNIT/Assert code for doing UNIT Testing for five Data Structure Lab program after converting the same to Java.
 - Selection Sort
 - Quick Sort
 - Stack
 - Queue
10. Create User Acceptance Test Cases for any existing popular website and compare results obtained with other student in the class

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA452E – Microcontroller Lab**.Microprocessor 8086:**

1. Introduction to TASM/MASM
2. Arithmetic operation – Multi byte addition and subtraction, multiplication and division- signed and unsigned arithmetic operation, ASCII-arithmetic operation.
3. Logic operations- Shift and rotate- converting packed BCD to unpacked BCD, BCD to ASCII conversion.
4. By using string operation and instruction prefix: Move block, reverse string, sorting, inserting, deleting,length of the string, string comparison.
5. DOS/BIOS programming: Reading keyboard (Buffered with and without echo)- Display characters, strings.

II. Interfacing:

1. 8259 – Interrupt Controller : Generate an Interrupt using 8259 timer.
2. 8279 – Keyboard display : Write a small program to display a string of Characters.
3. 8255 – PPI : Write ALP to generate sinusoidal wave using PPI.
4. 8251 – USART : Write a program to establish communication between two processors.

III. Microcontroller 8051

1. Reading and writing on a parallel port.
2. Timer in different modes.
3. Serial communication implementation.

MCA471 Seminar-II

Students have to select a topic related to the current trends and technologies in the field of Computer Science. They need to prepare the synopsis and detailed report in consultation with the faculty guide. Each Student has to give one hour presentation to their fellow classmates and to a panel of guides.

MCA531 - Computer Graphics with Open GL**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To familiarize the students with the concepts of computer graphics like line, circle drawing algorithms, transformations, clipping, projection, color models, curves. To make the students understand how to implement the computer graphics concepts using OpenGL.

Learning outcome

Upon successful completion of the course the students will be able to learn

- The concepts of Computer Graphics.
- Implementing the Graphics concepts using OpenGL.

Unit I.**(12)****Introduction to Computer Graphics**

Applications, Overview of Graphics Systems – Video display devices, Raster-scan systems, Graphics software, Introduction to OpenGL.

Graphics Output Primitives

Coordinate Reference Frames, Two-Dimensional frame in OpenGL, Point Functions, Line Functions, Line-Drawing Algorithms – DDA, Bresenhams, Curve Functions, Midpoint Circle Algorithm, and Display-window reshape function.

Self-Learn: Area filling, Display lists, Basic colors, Attribute functions.

Unit II.**(12)****Geometric Transformations**

Basic two-dimensional geometric transformations, Homogeneous Coordinates, Composite transformations, Geometric transformations in three-dimensional space, Translation, Rotation, scaling, composite three-dimensional transformations, OpenGL geometric transformation functions.

Self-Learn: Reflection, shear.

Unit III.**(12)****Illumination and Color Models**

Light sources, Basic illumination models, transparent surfaces, OpenGL illumination functions. Color Models, Standard primaries and chromaticity diagrams, RGB color model, HSV color model. OpenGL color functions.

Self-Learn: Ray-tracing and Texture mapping.

Unit IV.**(12)****Viewing**

Two-dimensional viewing pipeline, clipping window, Normalization and viewport transformations, 2D viewing functions, Clipping Algorithms – Line clipping – Cohen-Sutherland and Liang-Barsky Line clipping, polygon clipping – Sutherland-Hodgman algorithm.

Three-dimensional viewing concepts – Projections, Three-dimensional viewing pipeline, Projection transformation, Parallel and Perspective projection matrices. 3D viewing functions.

Self-Learn: Other clipping algorithms, Text clipping, and Projection derivations.

Unit V.

(12)

Three-dimensional Object Representations

Spline representations, Cubic spline interpolation methods, Bezier curves and B-Spline curves. OpenGL approximation-Spline functions.

Text book:

[1] D. Hearn, M. Pauline Baker, *Computer Graphics with OpenGL*. PHI, 3rd Edition, New Delhi, 2011.

Reference Books:

- [1] Foley, Vandan & Feiner, Hughes, *Computer Graphics Principles & Practice, in C*, Pearson Education (Singapore Pvt Ltd, Indian Branch, Delhi), 6th Indian Reprint 2001.
- [2] Richard S Wright, Jr. Michael Sweet, *Open GL Super Bible*, 2nd Edition.
- [3] Woo, Mason and Neider, Jackie, *Open GL Programming guide*.

MCA532 Artificial Intelligence**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To introduce basic theory and practical techniques in Artificial Intelligence. The course would provide emphasis to the principles and applications of Artificial Intelligence.

Learning outcome

Upon successful completion of the course the student would

- Understand what AI mean and the foundations of it.
- Understand those elements constituting problems and learn to solve it by various uninformed and informed (heuristics based) searching techniques
- Understand the formal method for representing the knowledge and the process of inference to derive new representations of the knowledge to deduce what to do
- Understand the notion of Planning, Game playing and NLP in AI and basic techniques in the classical systems

Unit I. (12)**Introduction**

Introduction to AI, The Foundations of AI, AI Technique -Tic-Tac-Toe.Problem characteristics, Production system characteristics, Production systems: 8-puzzle problem. Searching: Uniformed search strategies – Breadth first search, depth first search.

Unit II. (12)**Local Search Algorithms**

Generate and Test, Hill climbing, simulated annealing search, Constraint satisfaction problems, Greedy best first search, A* search, AO* search.

Unit III. (12)**Knowledge Representation**

Propositional logic – syntax and semantics, First order logic. Inference in first order logic, propositional Vs. first order inference, unification & lifts, Clausal form conversion, Forward chaining, Backward chaining, Resolution.

Unit IV. (12)**Game Playing**

Overview, Minimax algorithm, Alpha-Beta pruning, Additional Refinements.

Planning

Classical planning problem, STRIPS- basic process and working of system.

Unit V. (12)**Natural Language Processing**

Introduction, Syntax processing, Semantic Analysis, Pragmatic and Discourse Analysis.

Text Book:

- [1] E. Rich and K. Knight, *Artificial Intelligence*, 2nd Edition. New york: TMH, 2012, ISBN: 9780070087705
- [2] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*, 2nd Edition. Pearson Education, 2007.

Reference Books:

- [1] Eugene Charniak and Drew McDermott, *Introduction to Artificial Intelligence*, 2nd Edition. Singapore: Pearson Education, 2005.
- [2] George F Luger, *Artificial Intelligence Structures and Strategies for Complex Problem Solving*, 4th Edition. Singapore: Pearson Education, 2008, ISBN-13 9780321545893
- [3] N.L. Nilsson, *Artificial Intelligence: A New Synthesis*, 1st Edition. USA: Morgan Kaufmann, 2000.

MCA533 System Software**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

To provide basic knowledge of various system software to get deeper understanding of actual working of a computer system.

Learning outcome

- Understanding basics of system software.
- Understanding design approaches, techniques and tools for developing system software.

Unit I. (13)**Machine Structure and Evolution of a programming system**

Introduction to System Software, Components of System Software, Evolution of System Software, Assembler, Loader, Macros, Compilers, Simplified Instructional Computer: SIC machine architecture, SIC/XE machine architecture, SIC programming examples. Basic assembler functions (SIC assembler, algorithm and data structure), Machine dependent assembler features (Instruction formats and addressing modes, program relocation), Machine independent assembly features (Literals, Symbol defining statements, expressions, program blocks, control sections and program linking), Assembler design options (One pass assembler, multi pass assembler)

Unit II. (12)**Loaders and Linkers**

Basic loader functions (Design of an absolute loader, simple bootstrap loader), Machine dependent loader features (Relocation, program linking, algorithm and data structures for a linking loader), Machine independent loader features (Automatic library search, loader options), Loader design options (Linkage editor, dynamic linking, bootstrap loaders). Implementation Examples MS DOS linker, SunOS linker

Unit III. (11)**Macro Processor**

Macro Instructions, Features of a macro facility (Macro instruction arguments, Conditional macro expansion, Macro calls within macro, Macro instructions defining macros), Implementation (Two pass algorithm, Single pass algorithm), ANSI C Macro language

Unit IV. (12)

Basic compiler functions grammars, lexical analysis, syntactic analysis, code generation, machine dependent compiler features intermediate form of the program, machine dependent code optimization, Machine independent compiler features – structured variables, machine independent code optimization, storage allocation, block structured languages.

Unit V.**(12)**

Compiler design options, division into passes, interpreters, P code compilers, compiler-compilers, Design of Lexical Analyzer-Ex of a language specifying lexical analyzers.Example of parser generator.

Text books:

[1] A Beck, Leland, *System Software An Introduction to System Programming*, Addison-Wesley, 3rd Edition, 1997, Third Impression, 2008.

Reference Books:

[2] A.V.Aho and J.D. Ullman; “*Compilers-Principles, Techniques and Tools*” Addison Wesley, 3rd Indian Reprint, 2006.

MCA541A Software Architecture**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Course Objective**

To provide a sound technical exposure to the concepts, principles, methods, and best practices in software architecture and software design.

Learning Outcome

- An ability to conceptualize and coordinate designs, addressing technological aspects of architecture.
- An ability to produce "software architects" with sound knowledge and superior competence in building robust, scalable, and reliable software intensive systems in an extremely .
- An ability to recognize and analyze the Architecture.
- An ability to apply and integrate computer technology in design processes and products.

Unit I.**(11)****Introduction**

Architecture Business Cycle – Origin of an Architecture , Software Processes and Architectural Business Cycle, A good architecture, Software Architecture, What is & what it is not the software Architecture is, Other points of view, Architectural Pattern, Reference Models and Reference Architectures, The Importance of Software Architecture, Architectural structures & views, Case study in utilizing Architectural Structures.

Unit II.**(12)****Creating An Architecture**

Understanding the quality Attributes – Functionality and Architecture, Architecture and Quality Attributes, System Quality Attributes, Quality Attributes Scenarios in practice, Other System Quality Attributes, Business Qualities, Architecture Qualities.

Achieving Qualities – Introducing Tactics – Availability, Modifiability, Performance, Security, Testability, Usability, Relationships of Tactics to Architectural Patterns, Architectural Patterns and Style.

Unit III.**(11)****Design and Documentation**

Designing the Architecture – Architecture in the life cycle, Designing the Architecture, Forming the Team Structure, Creating the Skeletal System. Documenting Software Architectures, Uses of Architectural Documentation, Views, Choosing the relevant views, Documenting a view, Documentation across views.

Unit IV.**(13)****Analyzing Architecture**

ATAM (Architecture Tradeoff Analysis Method) – A comprehensive method for architecture evaluation, participants, outputs, phases of the ATAM, The Nightingale system - A case study in applying the ATAM.

CBAM (Cost Benefit Analysis Method) – A quantitative approach to architecture design decision making: Decision making context, basis for CBAM, Implementing CBAM, A Case Study – The NASA ECS project.

The World Wide Web – A case study in interoperability Relationship to the Architecture Business Cycle, Requirements & Quality, Architectural Solution, The evolution of web-based e-commerce architectures, Achieving quality goals, Architecture Business Cycle today.

Unit V.**(13)****Software Product Lines**

Reusing Architectural Assets – Overview – Successful working, Scope, Architectures and Difficulties in software product lines.

Celsuis Tech – A Case study in product Line development, Relationship to the Architecture Business Cycle, Requirements & Quality, Architectural Solution.

Building systems from off-the-shelf components – Impact of components on Architecture, Architectural mismatch, Component-based design as search, ASEILM example.

Text Book:

- [1] Len Bass, Paul Clements, Rick Kazman, *Software Architecture In Practice*, Pearson Education Asia , 3rd Edition, 2012.

Reference Books:

- [1] Sommerville, Ian, *Software Engineering*, Addison Wesley, 5th Edition, 2010.
- [2] Pressman S Roger, *Software Engineering*, Mc Graw Hill International Editions, 4th Edition, 2009.
- [3] Jeff Garland, Richard Anthony, *Large-Scale Software Architecture – A Practical Guide Using UML*, Wiley –dreamtech India Pvt.,Ltd., 2000.
- [4] Rumbaugh, James, *Object Oriented Modeling and design*, Pearson Education, New Delhi, 2005.

MCA541B Wireless and Mobile Networks**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

The goal is to make students familiar with the basic concepts and structure of modern wireless and mobile communication networks.

Learning outcome

Upon successful completion of the course

- Students will learn basic principles of wireless and mobile networks with focus on computer and data networks, Knowledge of basic protocols and interfaces.

Unit I. (12)

Wireless Telecommunications Systems and Networks: History and Evolution of Wireless Radio Systems, Development of Modern Telecommunications Infrastructure, Overview of Existing Network Infrastructure, Wireless Network Applications: Wireless Markets

Evolution and Deployment of Cellular Telephone Systems: Different Generations of Wireless Cellular Networks, 1G Cellular Systems, 2G Cellular Systems, 2.5G Cellular Systems, 3G Cellular Systems, 4G Cellular Systems and Beyond, Wireless Standards Organizations

Unit II. (12)

Common Cellular System Components: Common Cellular Network Components, Hardware and Software Views of the Cellular Network, 3G Cellular System Components, Cellular Component Identification, Cell establishment

Wireless Network Architecture and Operation: The Cellular Concept, Cell Fundamentals, Capacity Expansion Techniques, Mobility Management, Wireless Network Security.

Unit III. (13)

GSM and TDMA Technology: Introduction to GSM and TDMA, GSM Network and System Architecture, GSM Channel Concept, GSM Identities, GSM System Operations, GSM Infrastructure Communications.

Unit IV (12)

CDMA Technology: Introduction to CDMA, CDMA Network and System Architecture, CDMA Channel Concept, CDMA System Operations.

CDPD and Edge Data Networks: CDPD, GPRS, GPRS Networks, GPRS Network Details, GPRS Network Layout and Operation, GPRS Packet Data Transfer, GPRS Protocol Reference Model, GPRS Logical Channels, GPRS Physical Channels, GSM/GPRS/Edge Technology.

Unit V (11)

Wireless LAN/Wireless PANs/IEEE 802.15x: Introduction to wireless LAN 802.11X technologies, Evolution of Wireless LAN, Introduction to IEEE 802.15x Technologies,

Wireless PAN Applications and Architecture, Bluetooth, Introduction to Broadband wireless MAN, 802.16 technologies.

Text Book

- [1] Gary J Mullett. *Wireless Telecommunications Systems and Networks*, Clifton Park (N.Y.) : Thomson Delmar Learning, cop.2008

Reference Books

- [1] Raj Kamal, *Mobile Computing*, Oxford University Press, 2012.
[2] Stallings William, *Wireless Communications and Networks*, Pearson Education Asia, 2nd Edition, 2009.
[3] Theodore S Rappaport, *Wireless Communications: Principles and Practice*, Pearson Education Asia, 2nd Edition, 2009.
[4] Jochen Schiller, *Mobile Communication*, Addison-Wesley, 2nd Edition, 2011.

MCA541C Parallel Computing with OpenCL**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Perquisites**

Basics of computer architecture ,data structures and algorithms.

Objectives

The objective of this paper is to help the students to analyze existing algorithms and problems that has inherent parallelism. Also, this paper helps the students to understand and learn the OpenCL programming model for parallel programming.

Learning Outcome

Students will be able

- To implement and analyze algorithms in OpenCL
- To analyze the existing algorithms and problems that has inherent parallelism.

Unit I.**(12)****Introduction Parallel Computing**

Introduction to pipelining, parallelism –Instruction-level parallelism –Data-level parallelism – Task-level parallelism – Thread-level parallelism –Synchronization –Multi-core CPUs –GPU architecture –Memory organization in CPU and GPU – Architecture styles: SIMD, VLIW, vectorisation – GPU computing.

Unit II.**(12)****Introduction to OpenCL**

OpenCL basics: devices, 4 models, kernel basics – Organisation of an OpenCL program: kernel, platforms, devices, context, program, command queue, buffer read / write, release resources – Performance analysis of OpenCL programs – Examples in OpenCL .

Unit III.**(12)****OpenCL Extensions**

OpenCL memory spaces: Separate memory spaces from host, global/local/constant breakdown, explicit data movement – Threads and scheduling: Wavefronts, wide SIMD –Performance optimisation: memory space usage, memory coalescing, wavefront/warp occupancy – Extensions: device fission, double precision, Interop, zero copy – Timing Events and profiling: Timing using events, wall clock timing, profiling, debugging

Unit IV.**(12)****Parallel Algorithms on Sequences and Strings**

Sorting: Quick sort, Selection sort, Insertion sort, Counting sort - Merging: Batcher's Odd-Even merge, Halving merge – Medians: Quick select, Sample select –Searching: Hash tables – String Matching: Naive string matching

Unit V.**(12)****Parallel Algorithms on Graphs and Matrices**

Trees: Root-fix, Leaf-fix – shortest Path: Breadth first search – Matrix: addition, multiplication, inverse– Case Studies: Loop unrolling, Data reuse, Knapsack, image processing, Simple web plugin.

Text Book:

- [1] Benedict Gaster, Lee Howes, David R. Kaeli and Perhaad Mistry, “*Heterogeneous Computing with OpenCL*”, Elsevier Inc, August 2011

Reference Books:

- [1] Janusz Kowalik , Tadeusz Puzniakowski, “ *Using open CL programming Massively parallel computers* “, volume 21,IOS press, 2012
- [2] Aaftab Munshi, Benedict Gaster , Timothy G mattson, James Fung, Dan Ginsburg, “*OpenCL programming Guide*” , Addison-Wesley,2011
- [3] CLRS (T.H. CORMEN, C.E. LEISERSON, R.L. RIVEST, C. STEIN), “*Introduction To Algorithms*”, 2nd/3rd Edition, Prentice Hall India, 2009.
- [4] D. Kirk and W. Hwu, “*Programming Massively Parallel Processors*”, Morgan Kaufmann, ISBN: 978-0-12-381472-2.
- [5] SCandAL Project, Carnegie Mellon University, “*A Library of Parallel Algorithms*”, <http://www.cs.cmu.edu/~scandal/nesl/algorithms.html>

Web Source:

- [1] OpenCL University Kit,
http://developer.amd.com/downloads/opencl_univ_kit_1.0.zip

MCA541D Machine Learning*Total teaching Hours/Semester:60**No of Lecture Hours/Week: 04***Objective**

To acquire basic knowledge in machine learning techniques and learn to apply the techniques in the area of pattern recognition and data analytics.

Learning outcome

Upon completion of the course students will be able to:

- Understand the basic principles of machine learning techniques.
- Understand the supervised and unsupervised machine learning algorithms.
- Choose appropriate techniques for real time problems.

Unit I.**(12)**

Introduction: Machine Learning, types of machine learning, examples.

Supervised Learning: Learning class from examples, VC dimension, PAC learning, noise, learning multiple classes, regression, model selection and generalization, dimensions of a supervised learning algorithm.

Parametric Methods: Introduction, maximum likelihood estimation, evaluating estimator, Bayes' estimator, parametric classification.

Unit II.**(12)**

Dimensionality reduction: Introduction, subset selection, principal component analysis, factor analysis, multidimensional scaling, linear discriminant analysis.

Clustering: Introduction, mixture densities, k-means clustering, expectation-maximization algorithm, hierarchical clustering, choosing the number of clusters.

Non-parametric: Introduction, non-parametric density estimation, non-parametric classification.

Unit III.**(10)**

Decision Trees: Introduction, univariate trees, pruning, rule extraction from trees, learning rules from data.

Multilayer perceptron: Introduction, training a perceptron, learning Boolean functions, multilayer perceptron, backpropagation algorithm, training procedures.

Unit IV.**(14)**

Kernel Machines: Introduction, optimal separating hyperplane, v-SVM, kernel tricks, vertical kernel, defining kernel, multiclass kernel machines, one-class kernel machines.

Bayesian Estimation: Introduction, estimating the parameter of a distribution, Bayesian estimation, Gaussian processes.

Hidden Markov Models: Introduction, discrete Markov processes, hidden Markov models, basic problems of HMM, evaluation problem, finding the state sequence, learning model parameters, continuous observations, HMM with inputs, model selection with HMM.

Unit V.**(12)**

Graphical Models: Introduction, canonical cases for conditional independence, d-separation, Belief propagation, undirected graph: Markov random field.

Reinforcement Learning: Introduction, single state case, elements of reinforcement learning, temporal difference learning, generalization, partially observed state.

Text Books:

[1] E. Alpaydin, *Introduction to Machine Learning*. 2nd MIT Press, 2009.

Reference Books:

[1] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[2] P. Harrington, *Machine Learning in Action*. Manning Publications, 2012

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2011.

[4] S. Marsland, *Machine Learning: An Algorithmic Perspective*. 1st Ed. Chapman and Hall, 2009

[5] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997

MCA541E Embedded Programming and RTOS**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

This course provides an introduction to embedded real-time operating systems. Topics covered include general embedded systems concepts, general embedded software development, real-time operating systems concepts.

Learning Outcome

Students who successfully complete this course will have :

- Familiarity with many of the issues involved with embedded systems.
- Familiarity with key Real-Time Operating System terms and concepts.
- Ability to program an embedded system with tasks and executive.
- Understanding and ability to use tools to build an embedded real-time system.
- Ability to specify, design and implement a small embedded system.
- Ability to present design information effectively in the forms of technical reports and oral presentations.

Unit I.**Embedded Programming using C****(11)**

Intrinsic routines, Library files, Buffer manipulation, Character conversion and classification, Data conversion, Memory allocation, Stream input and output, String manipulation, Variable length argument lists, Compiler Language Extensions(Data Types, Memory Types, Memory models, Pointers, Interrupt Procedure)

Unit II.**Real time Operating system:****(11)**

Typical Real time Applications & Hard versus Real time Applications:

Digital control, High level controls, Signal processing, Other Real time applications, Jobs and processors, Release times, Deadlines and Timing constraints, Hard and Soft Timing constraints, Hard Real time systems, Soft Real time systems:

A reference model of Real time systems:

Processors & Resources, Temporal parameters of real time Workload, Periodic Task model, Precedence constraints and data dependency, Other types of dependencies, Functional parameters, Resource parameters for jobs and parameters of Resources, Scheduling hierarchy

Unit III.**Operating systems****(12)**

Overview, Threads & tasks, The Kernel, Time services and scheduling mechanism, Time services: clocks & time, Resolution, High resolution, Timers & Timers functions, Asynchronous timer functions, Synchronous timer functions, Timer resolution, Periodic time interrupts, one shot Timer interrupts, Timer accuracy, Release time jitters of periodic tasks

Scheduling mechanisms: Fixed priority Scheduling, EDI scheduling, preemption lock, Aperiodic thread scheduling, monitoring processor time consumption, Tracking busy intervals, Hook for user level Implementation, static configuration , Release Guard mechanism

Unit IV.

Other basic operating system functions

(13)

Communication and synchronization, Event notification and software Interrupts, memory management, I/O and networking

Processor Reserves and Resource kernel:

Resource model and Reservation types, Application program interface and SSP structures

Open system architecture

Objectives & alternatives, Two level scheduler, server maintenance, Sufficient schedulability condition and acceptance test, Scheduling overhead and processor utilization, service Provider structure and real time API Functions

Unit V.

Capabilities of commercial Real time Operating

(13)

LynxOS, pSOSystem, QNX/Neutrino, VRTX, VxWorks

Predictability of General purpose operating systems

Windows-NT Operating system: scheduling, limited priority levels, jobs, jobs scheduling classes, User level NPCS, ceiling priority protocol, deferred procedure calls

Real Time extension of Linux Operating system:

Important features, scheduling, clock and timer resolution, threads, UTIME High resolution Time service

Text Books:

[1] Liu, Jane S. *Real time systems*, Pearson education, 2006

[2] Mukhi, Vijay. *The 'C' Odyssey UNIX*, BPB publications, 2004

[3] JeeSe Russell, Ronald Cohn, Real-Time Operating System, Book on Demand Ltd, 2012

Reference Book:

[1] WilmShurst , Tim, *An Introduction to the Design of small scale embedded systems*, Palgrave Macmillan, 2001

MCA542A Information Retrieval and Web Mining**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

The main objective of the course is aimed at an entry level study of information retrieval and web mining techniques. It is about how to find relevant information and subsequently extract meaningful patterns out of it. While the basic theories and mathematical models of information retrieval and web mining are covered, the course is primarily focused on practical algorithms of textual document indexing, relevance ranking, web usage mining, text analytics, as well as their performance evaluations.

Learning Outcomes

Upon successful completion of the course, Students are expected to master both the theoretical and practical aspects of information retrieval and web mining. More specifically,

- The student will understand the basic concepts and processes of information retrieval systems and data mining techniques.
- The common algorithms and techniques for information retrieval (document indexing and retrieval, query processing, etc).
- The quantitative evaluation methods for the IR systems and web mining techniques.
- The popular probabilistic retrieval methods and ranking principle.

Unit I.**(10)**

Introduction: Introduction to Data mining. Relationship to machine learning. Summarization and feature extraction. **Data Preprocessing:** Introduction to preprocessing. Data summarization. Data cleaning. Data integration, Data transformation. Data cube aggregation, attribute subset selection, Dimensionality reduction, Numerosity reduction. Data Discretization, Concept Hierarchy generation.

Unit II**(12)**

Introduction to Information Retrieval: Inverted indices and Boolean queries. Query optimization. The nature of unstructured and semi-structured text. The term vocabulary and posting lists. Text encoding: tokenization, stemming, lemmatization, stop words, phrases. Optimizing indices with skip lists. Proximity and phrase queries. Positional indices. Dictionaries and tolerant retrieval. Dictionary data structures. Wild-card queries, permuterm indices, n-gram indices. Spelling correction and synonyms: edit distance, soundex, language detection.

Index construction. Postings size estimation, sort-based indexing, dynamic indexing, positional indexes, n-gram indexes, distributed indexing

Unit III.**(12)**

Scoring, term weighting, and the vector space model. Parametric or fielded search. Document zones. The vector space retrieval model. tf.idf weighting. The cosine measure. Scoring

documents. Map Reduce: Distributed file systems, Map and reduce tasks. Algorithms that use map-reduce: Matrix vector multiplication, Relational algebra operations.

Mining Frequent Patterns and Associations: Near-neighbor search, Collaborative filtering, Shingling. Min-hashing and locality sensitive hashing.

Unit IV

(13)

The stream data model, examples of stream sources and queries, sampling data in a stream. Filtering streams, bloom filters, counting distinct elements in a stream. Market-Basket model, Association rules. A-priori algorithm. Classification: Introduction to text classification. Naïve Baye's models. Spam filtering. K nearest neighbors, Decision boundaries, vector space classification using centroids. Comparative results. Support vector machine classifiers. Kernel function. Evaluation of classification. Micro-and macro-averaging. Learning rankings.

Unit V.

(13)

Clustering: Introduction to the problem. Partitioning methods: K-means clustering; Hierarchical clustering. Latent semantic indexing (LSI). Applications to clustering and to information retrieval. Web Mining: Introduction to web . Web search overview, web structure, the user, paid placement, search engine optimization/spam. Web measurement. Crawling and web indexes. Near-duplicate detection. Link analysis. Web as a graph. PageRank. Machine learning techniques for ranking.

Text Book:

- [1] C. Manning, P. Raghavan, and H. Schütze, .“ *Introduction to Information Retrieval* “,Cambridge University Press, 2008.
- [2] Anand Rajaraman and Jeffery D.ullman, ”*Mining the Massive*”,Cambrize University Press, 2008.

Reference Books:

- [1] Data,Bing Liu, .”*Web Data Minig,Exploring Hyperlinks,contents and usage*”,2nd Edition, July 2011,Springer.
- [2] K.P Soman, Shyam diwakar,VAjay, *Insight into Data Mining – Theory and Practice*, 6th print, PHI India, 2012
- [2] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, 2006, Morgan Kaufmann Publishers, San Francisco, USA.

MCA542B Database Administration**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

The course provides insight on the administrative tasks, their implementation and effective usage of tools.

Learning outcome

Upon successful completion of the course students would

- Have sound knowledge of the administrative tasks
- Install ,configure Oracle
- Database connectivity and User management
- Basic Networking and security implementation.

Unit I.**(12)****Introduction and Oracle 11g Architecture**

Introduction: General Definition of DBA and Security, System Management & Database Design Roles of DBA – DBA Job Classification. Types of Databases: Online Transaction Processing System and Decision Support System Databases, Development, Test & Production Databases. Daily Routine of a DBA.

Architecture: Database Structures- Logical & Physical, Trace Files, Data Files & Tablespaces, Oracle Managed Files. Processes- Interaction between User & Oracle Processes, The Server Process, Background Processes. Memory Structures- SGA, PGA. Oracle Transactions- Anatomy of SQL Transactions. Data Consistency & Concurrency- Database Writer & Write Ahead Protocol, The System Change Number, Undo Management. Backup and Recovery Architecture-User managed, RMAN, Flashback Techniques. Data Dictionary and Dynamic Performance Views- Data Dictionary, V\$ views, The Oracle Optimizer. Oracle utilities, Automatic Database Management, Advisory Framework.

Unit II.**(12)****Database Installation and Creation**

Installing Oracle 11g: Following OFA, System and Owners Pre-Installation Tasks, Installing Software, System Administrator and Oracle Owner's Post-Installation Tasks, Uninstalling Oracle 10g.

Database Creation: Creating SPFILE and pfile, Initialization Parameters, Creating a new Database, Using SPFILE, Starting up and Shutting Down Database.

Unit III.**(13)****Database Connectivity and Networking, User Management and Security**

Database Connectivity and Networking: Working of Oracle Network – instance names, global database names, connect descriptors, identifiers and strings, Establishing Connectivity, Oracle Client, Installing the Client, Naming and Connectivity – Local, Easy connect, External and Directory naming methods.

Managing Users: Creating, altering and dropping users, Creating user Profiles & Resources, Database Resource Manager, Controlling Access to Data – Roles, Privileges and using Views,

Stored Procedures to Manage Privileges, Auditing Database – Standard Auditing, Authentication – Database, External, Centralized user and Proxy Authentication. Database Security Do's & Don'ts-User Accounts, Passwords, OS authentication, Auditing Database, Granting Appropriate Privileges, Permissions, Application Security.

Unit III.**(11)****Data Loading**

Loading and Transforming Data: Overview of extraction loading and Transformation, Loading Data-Using the SQL Loader Utility, Using External Tables to Load Data. Overview of Common Techniques used for Transforming Data. Data Pump Technology: Introduction, Benefits, Uses and Components of Data Pump. Access method, Data Pump Files, Privileges, Mechanics of Data Pump Job.

Unit IV.**(12)****Backup, Recovery & Database Performance Tuning**

Backing Up Oracle Databases: Backup Terms, Guidelines, Strategies, Examining Flash Recovery Area – benefits of Flash recovery Area, Looking into Flash Recovery Area, Setting size of Flash Recovery Area Creating Flash Recovery Area, Backing up Flash Recovery Area, RMAN – Benefits, Architecture, Connecting to RMAN.

SQL Query Optimization: Approach to Performance Tuning, Optimizing Oracle Query Processing, Cost-based Optimizer, Drawbacks of CBO. SQL Performance Tuning Tools – EXPLAIN PLAN, Auto trace, SQL Trace and TKPROF.

Tuning the instance: Introduction, Automatic Tuning vs. Dynamic Views. Tuning Oracle Memory: Tuning Shared Pool – Library Cache, Dictionary Cache, Hard vs. Soft Parsing, Sizing Shared Pool, Tuning Buffer Cache – Sizing buffer Cache, Multiple pools for Buffer Cache, Tuning Large, Streams and Java Pools. Tuning PGA Memory – Automatic PGA Memory Management.

Text Book:

- [1] Alapati, Sam R., *Expert Oracle Database 11g Administration*, Springer India Pvt. Ltd., 2009.

Reference Books:

- [1] Alapati, Sam R., *Expert Oracle Database 10g Administration*, Springer India Pvt. Ltd., 2008.
[2] Kyte, Thomas, *Expert Oracle, Oracle Press Publication*, Signature Edition, 2005.

MCA542C Cloud Computing**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objective**

Cloud computing has become a great solution for providing a flexible, on-demand, and dynamically scalable computing infrastructure for many applications. Cloud computing presents a significant technology trend. The course aims at familiarizing with the basic concepts of cloud computing and its applications.

Learning outcome

Upon successful completion of the course

- Understand the common terms and definitions of virtualization and cloud computing and be able to give examples.
- Comprehend the technical capabilities and business benefits of virtualization and cloud computing.
- Describe the landscape of different types of virtualization and understand the different types of clouds.
- Illustrate how key application features can be delivered more easily on virtual infrastructures.

Unit I.**(12)**

Cloud Computing Fundamentals - Introduction- Layers of cloud computing – Types of Cloud computing – Cloud computing versus cloud services – Enabling Services – Virtualization – Web Service and Service Oriented Architecture- Service flow and Work flows- Web 2.0 and Mash up - Cloud Computing Features – Cloud Computing standards – Cloud Computing security - Cloud Computing Platforms – Pricing- Cloud Computing Components and Their Vendors - Example of Web Application Deployment - Cloud Computing Challenges – Performance- Security and Privacy – Control- Bandwidth Costs – Reliability.

Unit II.**(12)**

Cloud Computing Technologies and Applications - Cloud Computing: IT as a Service - Cloud Computing Security - Cloud Computing Model Application Methodology - Cloud Computing Strategy Planning Phase - Cloud Computing Deployment Phase - Cloud Computing in Development/Test - Cloud-Based High Performance Computing Clusters - Use Cases of Cloud Computing - Case Study: Cloud as Infrastructure for an Internet Data Center (IDC) - The Bottleneck on IDC Development - Cloud Computing Provides IDC with a New Infrastructure Solution - The Value of Cloud Computing for IDC Service Providers - The Value Brought by Cloud Computing for IDC Users - An IDC Cloud Example .

Unit III.**(12)**

Cloud Types and services - Introduction - Cloud Types - Public Cloud – Private Cloud - Hybrid Cloud – Community Cloud - Cloud Services and Cloud Roles – Infrastructure as Service - Amazon Elastic Compute Cloud (EC2) - Amazon Simple Storage Service (S3) -

Platform as a Service - Google App Engine - Microsoft Azure - Software as a Service - Google Apps.

Service Scalability over the Cloud – Introduction – Foundations - History on Enterprise IT Services - Warehouse-Scale Computers- Grids and Clouds - Application Scalability - Automating Scalability - Scalable Architectures - General Cloud Architectures for Scaling.

Unit IV.

(12)

Data-Intensive Technologies for Cloud Computing - Introduction - Data-Intensive Computing Applications- Data-Parallelism- The Data Gap - Characteristics of Data-Intensive Computing Systems - Processing Approach- Common characteristics- Grid Computing - Applicability to Cloud Computing - Data-Intensive System Architectures - Google MapReduce - Hadoop - LexisNexis HPCC - Architecture Comparison.

Unit V.

(12)

Cloud Computing with Titans –Google – Google App Engine – EMC – NetApp- Microsoft – Amazon – IBM- Partnerships.

Accessing the cloud – Platforms – Web Applications- Web API- Web Browsers.

Developing Applications – Google- Microsoft – Development - Google App Engine – Application Management

Text Book:

[1] *Handbook of Cloud Computing*, Springer Science + Business Media, LLC 2010

[2] Anthony TVelte, Toby JVelte and Robert Elsenpeter, *Cloud Computing –A Practical Approach*, Tata McGraw Hill Education Pvt Ltd, 2010

Reference Books:

[1] Syed A.Ahson and Mohammed Ilyas, *Cloud Computing and Software Services : Theory and Techniques*, CRC Press, Taylor and Francis Group, 2010

[2] Judith Hurwitz, Robin Bloor, Marcia Kaufman and Fern Halper, *Cloud Computing for Dummies*. Wiley- India edition,2010

[3] Ronald L. Krutz and Russell Dean Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Wiley Publishing, Inc.,2012

[4] Barrie Sosinky, *Cloud Computing : Bible*, 1st edition, Wiley Publishing, Inc.,2011

[5] Ronald L. Krutz and Russell Dean Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Wiley Publishing, Inc.,2012

MCA542D Bioinformatics**Total teaching Hours/Semester: 60****No of Lecture Hours/Week: 04****Objectives**

To develop the skills of the students in Bioinformatics, develop knowledge on DNA mapping and sequencing, acquire knowledge on Database Algorithm and alignment Techniques, understand the nature of biological data and need for Biological databases, understand and explore the major bio molecular sequence databases (organization and contents) and their respective search engines and database searches, know the application of software analysis tools to sequence tools.

Learning outcome:

At the end of this course, the students would have learnt about tools used in Bioinformatics and how to use them. This will facilitate the students to undertake projects in the modern biology.

Unit I. (11)

Bioinformatics : An Introduction, Definition, Applications, Major Databases in Bioinformatics, Data Management and Analysis, Central Dogma of Molecular Biology.

Information search and retrieval : Introduction, tools for web search , Data retrieval tools, Data mining and biological databases.

Unit II. (13)

Genome Analysis and Gene Mapping : Introduction, Genome Analysis, Gene Mapping, Sequence Assembly Problem, Genetic Mapping and Linkage Analysis, Physical Maps, Genome sequencing, Application of Genetic maps, Human Genome project(HGP).

Alignment of pairs of sequences : Introduction, Biological motivation of Alignment problems, Methods of sequence Alignments, using scoring matrices, measuring sequence detection efficiency.

Unit III. (12)

Alignment of multiple sequences: Methods of multiple sequence alignment, Evaluation of multiple alignments, Applications of multiple alignments.

Tools for similarity search and sequence alignment : Introduction , FASTA, BLAST, Filtering and Gapped BLAST, FASTA and BLAST Algorithms comparison.

Unit IV. (13)

Perl for Bioinformatics : Sequences and strings- Representing Sequence data, Program to store DNA Sequence, Concatenating DNA fragments, Transcription : DNA to RNA, Perl documentation, Calculating the Reverse Complement in Perl, Proteins, Files and Arrays, Reading proteins in files, Arrays, Scalar and List Context.

Motifs and loops – Flow control, code layout, Finding motifs, Counting nucleotides, Exploding strings into arrays, Operation on strings, Writing to files - **Subroutines and Bugs** – Introduction, Scoping, Command Line Arguments, Passing data to subroutines, Modules and Libraries of Subroutines, Fixing bugs in the code.

Unit V.

(11)

Genetic code: Hashes, Data structures and Algorithms for Biology, Translating DNA into proteins, Reading DNA from files in FASTA Format, Reading Frames.

GenBank: GenBank files, GenBank libraries, Separating Sequence and annotation, Parsing annotations, Indexing GenBank with DBM.

Protein data bank: Files and folders, PDB files, Parsing PDB files, Controlling other programs.

Text Books

[1] S. C. Rastogi, Namita Mendirata, Parag Rastogi, *Bioinformatics: Methods and Applications*, 3rd Edition, PHI, 2011.

[2] Tisdall James, *Beginning Perl for Bioinformatics*, First edition, Shroff Publishers (O'Reilly), 2005

Reference Books

[1] T.K. Attwood and D.J. Perry Smith., *Introduction to Bioinformatics*, Pearson, 2011

[2] Arthur M. Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2003

[3] Irfan Ali Khan, Aditya Khanum, "Fundamentals of Bioinformatics", Ukaaz publications, 2003

[4] Harshawardhan P Bal, *Perl Programming for Bioinformatics*, Tata McGraw-Hill Education private limited, 2009

Web Sources:

- www.ncbi.nlm.nih.gov/tools
- www.ebi.ac.uk/tools

MCA542E Principles of User Interface Design*Total teaching Hours/Semester: 60**No of Lecture Hours/Week: 04***Objective**

The objective of this course is for students to learn how to design, prototype and evaluate user interfaces to effectively browse and search systems by examining what research has uncovered, what developers have produced, and how people perform information tasks.

Learning Outcome

- Learn the main concepts in human computer interaction.
- Learn basic user interface principles through practice.
- Learn about cognitive and perceptual abilities and constraints that impact information use.
- Learn about human information processing and how it is applied to the design of user interfaces.
- Learn to conduct user and task analysis specifically for information retrieval tasks.
- Learn to design and prototype user interfaces.
- Prepare for further training and research in this area.
- Be familiar with research issues in user interface design.

Unit I.**User-Interface****(12)**

Goals of User-Interface Design, Human factors in user interface design, Theories, Principles, and Guidelines, Goals of Systems Engineering, Accommodation of Human Diversity, Goals for Our Profession, High Level Theories, Object-Action Interface model, Principle 1: Recognize the Diversity, Principle 2: Use the Eight Golden Rules of Interface Design, Principle 3: Prevent Errors, Guidelines for Data Display, Guidelines for Data Entry, Balance of automation and Human Control, Practitioner's Summary, Researcher's Agenda.

Management Issues

Introduction, Organizational; Design to Support Usability, The three Pillars of Design, Development Methodologies, Ethnographic Observation, Participatory Design, Scenario Development, Social Impact Statement for Early Design Review, Legal issues, Expert Reviews, Usability, testing and Laboratories, Surveys, Acceptance tests, Evaluation During Active Use, Controlled Psychologically Oriented Experiments, Practitioner's Summary, Researcher's agenda.

Unit II.**(12)**

Tools Environment, and Menus: Introduction, Specification Methods; Interface-Building Tools, Evaluation and critiquing Tools. Direct Manipulation and virtual Environments: Introduction, Examples of Direct manipulation systems, Explanations of Direct manipulation, Visual Thinking and Icons, Direct Manipulation Programming, Home Automation, Remote Direct manipulation, Virtual Environments Menus: Task-Related Organization, Item Presentation Sequence, Response Time and Display Rate, Fast Movement through Menus, Menu Layout, From Fillin, Dialog boxes, Command-Organization strategies, The Benefits of

Structure, Naming and Abbreviations, Command Menus, Natural Language in Computing, Practitioners Summary, Researcher's Agenda.

Unit III.

(12)

Interaction Devices, Response Times, Styles and Manuals: Interaction Devices, Introduction, Keyboards and Function Keys, Pointing Devices, speech Recognition, Digitization, and Generation, Image and Video displays, Printers. Response Time and Display Rate: Theoretical; Foundations, Exceptions and attitudes, User Productivity, variability, Presentation Styles and Manuals: Introduction, Error messages, Nonanthropomorphic Design, Color of Manuals, Help: Reading From paper Versus from Displays, Preparation of Printed manuals, Preparation of Online Facilities, Practitioner's Summary, Researcher's Agend.

Unit IV.

(12)

Multiple-Windows, Computer-Supported Cooperative work, Information's search and www Multiple-Windows Strategies: Introduction, Individual-Window Design, Multiple-window Design, Coordination by Tightly-Coupled Windows, Image Browsing and Tightly-Coupled Windows, Personal Role Management and Elastic Windows Computer-Supported Cooperative Work; Introduction, Goals of Cooperation, Asynchronous Interactions: Different Time, Different Place, Synchronous Distributed: Different Place, Same Time, Face to Face: Same Place, Same Time, Applying CSCW to Education.

Unit V.

(12)

Information Search and Visualization: Introduction, Database Query And Phrase Search in Textual Documents, Multimedia Document Searches, Information Visualization, Advanced Filtering. Hypermedia and the World wide Web: Introduction, Hypertext and Hypermedia, World Wide Web, Genres and Goals and Designers, Users and Their Tasks, Object Action Interface Model for Web Site Design, Practitioner's summary, Researcher's Agenda.

Text Book:

- [1] Ben Shneiderman, *Designing the User Interface*, Pearson Education, 5th Edition, 2010
- [2] Wilber O Galitz, *An Introduction to GUI Design Principles and Techniques*, John-Wiley & Sons, 2007

Reference Books:

- [1] Jeff Johnson, *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*, Morgan Kaufmann, 1st Edition, 2010.
- [2] Alan J Dix et al, *Human-Computer Interaction*, Pearson, 2009.

MCA551 Computer Graphics Lab***Total Hours/Semester: 60******No of Hours/Week: 04***

1. Write a program to draw objects using OpenGL output primitive functions.
2. Write a program to draw objects using Line-Attribute, Fill-Area Attribute, Color functions of OpenGL.
3. Write a program to implement 2D Geometric Transformations.
4. Write a program to implement 3D Geometric Transformations.
5. Write a program to create a scene using OpenGL illumination functions.
6. Write a program to create a pattern for an object using OpenGL Texture Functions.
7. Write a program to implement Clipping.
8. Write a program to implement Projection.
9. Write a program to draw splines .
10. Write a program to draw an object surface.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

MCA552 Computer Networks Project Lab

Total Hours/Semester: 60

No of Hours/Week: 04

1. Network Security : Cryptography, Steganography, Digital Signature, Firewall.
2. Network Communication: IPC, IRDA, Radio wave, Bluetooth, Wi-Fi, Mobile Streaming, Client-server, Master-Slave.
3. Network Monitoring.
4. Ad-hoc networking, Remote login & Control
5. Application: e-governance
6. Implementation of different network protocols (SIP, RTP, RTCP, VOIP, SNMP, ARP, RARP and so on.

MCA553 Specialization Project Lab

Total Hours/Semester: 60

No of Hours/Week: 04

MINI PROJECT : Project based on previous semester's electives.

MCA651 Industry Project
(Sixth Semester Project)

No of Hours/Week: 30

It is a full time project to be taken up either in the industry or in an R&D organization.