Christ University

Hosur Road, Bangalore - 560029

Department of Computer Science

Syllabus for
MCA-2012 June onwards

Credit System for MCA

Subject	**Semesters- No. of hours (credits)**						***Hours***	***Credits***	***Marks***
	I	***II***	***III***	***IV***	***V***	***VI***			
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)**	**35(24)**	**34(22)**	**32(21)**	**32(21)**	**30(06)**	**196**	**117**	**4400**
Marks	**800**	**850**	**850**	**800**	**800**	**300**			
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	Discrete Mathematical structures	04	03	100
MCA135	Probability and Statistics	04	03	100
MCA136	Human Resource Management	04	03	100
MCA151	C 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	Software Engineering	04	03	100
MCA234	Relational Database Management System	04	03	100
MCA235	Operating Systems	04	03	100
MCA236	Accounting & Financial Management	04	03	100
MCA251	Assembly Language Programming Lab	04	02	100
MCA252	C++ Lab	04	02	100
MCA271	Seminar-1 (only CIA)	02	01	50
HOL02	Holistic Education	01	01	-
Total		**35**	**24**	**850**

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 in to five major units.
- From each major unit 2 full questions with internal choice (to select one) has to be set.
- Each full question carries 20 marks.
- There can be maximum of 3 sub divisions in each full question.

MCA131 Programming using C

Total teaching Hours/Semester: 60 ***No of Lecture Hours/Week: 04***

Objective

- Understand the concept of a C program like 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 (sscanf & sprintf)- 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. (12)

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.

Assembly language support

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

Unit V. (12)

Graphics Programming

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

Mouse Programming

Drawing with the mouse – Building mouse cursors – Freehand drawing using mouse – menus using mouse.

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
2. Gottfried Byron, *Programming with C*, Tata McGraw Hill
3. Kanetkar Yeshwant, *Understanding Pointers in C*, BPB publications, 4th edition, 2008
4. 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. **(10)**

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. **(14)**

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. (10)
MySql

Introduction to MySQL, CRUD - Select statements, Creating Database/Tables, Inserting values, updating and Deleting, PHP with MySQL, Creating Connection, Selecting Database, Perform Databse (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. (14)
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. Nicholas C Zakas , *Professional Java Script for Web Developers*, Wiley Publishing (Wrox) , II Edition , 2009
3. Matt Doyle , *Beginning PHP 5.3*, Willey Publishing , 2010

Reference Books:

1. Joseph W Lowery , *HTML 5 24 Hour Trainer* **,** *Wiley Publishing* **,** *2011 (Wrox)*
2. *D*avid 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 completion of this 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, Standard forms, Karnaugh Map, Map Simplifiation (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, 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.

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.

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 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, 5thedition, 2004.
2. B. Kolman, R. C. Busby and S. C. Ross, *Discrete Mathematical Structures,* Pearson Education, 5thedition, 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**

Part–E: Five (5) questions to be answered out of seven (7), **Marks: 5 x 6 = 30**

Each carrying five (6) marks.

Four(4) Questions to be asked from Unit-IV

Three(3) questions to be asked from Unit-V

Total Marks = 100

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- Arithmatic 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)**

Probabilty 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.
- To 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. **(8)**

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. **(8)**

Job Analysis: Job Description and Job Specification.

Human Resource Planning

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

Personnel Administration Data Systems (PADS)

Three types of PADS: Personnel Reporting Systems, Personnel Action Systems, Fringe Benefit Administration Systems.

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. (8)

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 VI. (12)

Labor Management Relations

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

Trade Unions

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
4. Edwin B Flippo, Human Resource Management, Tata McGraw Hill, 10th Edition, 2011.
5. 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 x 5 = 10
Part–B: Five questions to be answered out of six	5 x 5 = 25
Part–C: Three questions to be answered out of four	15 x 3 = 45
Part–D: Case study (compulsory)	20 x 1 = 20

Total Marks = 100

MCA151 C Lab

Total Hours/Semester: 60 ***No of Hours/Week: 04***

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 schemes, passing of arrays
7. Demonstration of recursion
8. Demonstration of various string operations
9. Implementation of the storage types
10. Demonstration of pointer operations.
11. Demonstration of macro processing.
12. Implementation of structures and array of structures
13. Implementation of Union.
14. Implementation of pointers to structures and unions.
15. Demonstration of dynamic allocation of memory
16. Demonstration of bitwise operations.
17. Demonstration of various Text file operations.
18. Demonstration of various Binary file operations.
19. Demonstration of various fixed shapes with some animation
20. Demonstration of different graphics functions
21. Demonstration of non common figures
22. Demonstration of mouse programming

QUESTION PAPER PATTERN

Two questions will be selected by the examiners (one from the list and one from outside the list). 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 (one from the list and one from outside the list). 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 introduce basic postulates of Boolean algebra, methods for simplifying Boolean expressions using laws and K-maps.
- It outlines the formal procedures for the analysis and design of combinational circuits like multiplexers, demultiplexers, decoders, encodes etc. and sequential circuits like various types of flip-flops, shift registers and synchronous and asynchronous counters.
- The course also focuses on the basic element, function and architecture of 8085 microprocessor and its operations .Finally it also touches upon programming techniques in developing the assembly language program for microprocessor application.

Learning Outcome

After the completion of the subject, students should be able to :

- Understand the number systems and conversions form one to other.
- Be able to use Boolean algebra, and methods to simplify the expression using K-maps and design logic circuits.
- Understand the concepts and working of sequential circuits and combinational circuits, Identify the basic element, functions and architecture of 8085 microprocessor.
- Finally learn the instruction set of 8085 and code in the assembly language program to develop the microprocessor based application.

Unit I. **(11)**

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.

Architecture of 8085 MPU

ALU, Timing and Control Unit, Instruction Register and Decoder, Register Array .

Unit II. **(14)**

8085 Machine cycles and bus Timings

Opcode Fetch Machine cycle, Memory Read Machine cycle, Memory Write Machine Cycle, IO Read Machine cycle, IO Write Machine Cycle, Calculation of execution time for a program with examples

Introduction to 8085 programming

Instructions DATA Format and Storage, Addressing Modes, Instruction Classification – Data Transfer Instructions, Arithmetic Instructions, Logic Instructions, Branching Instructions, Machine Control Instructions, Assembly Language programs: Binary and BCD addition of two 32 bit numbers, Binary and BCD subtraction of 16 bit number, shifting 8 bit number by 1or 2 bit.

Unit III. **(12)**

8085 Programming Techniques

Counters and Time Delays – Hexadecimal counter, 0-9 counter, Stack and Subroutines, Restart Conditional Call and Return Instructions, Advanced Subroutine Concept, Assembly Language programs: Multiplication and division of 8 bit numbers.

Interrupts

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

Unit IV. **(12)**

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, Interfacing Key Board & Seven Segment Display.

Unit V. **(11)**

Introduction to Microcontroller

Introduction, 8051 Microcontroller Hardware, Input/Output Pins, Ports, and Circuits, External Memory, Serial Data Input/Output.

Text Books:

1. Ramesh.S.Goankar ,*Microprocessor Architecture, Programming & Applications With 8085* **,** 5th edition – Penram International – 2011.
2. Kenneth J. Ayala, Dhananjay V. Gadre, *The 8051 Microcontroller & Embedded System using Assembly & C*,CENGAGE Learning(India edition), 2010.

Reference Books:

1. Soumitra Kumar Mandal, *Microprocessors and Microcontrollers Architecture, Programming and Interfacing using 8085, 8086 and 8051*, 1st edition. TMH, 2011.
2. Krishna Kant, *Microprocessors And Microcontrollers: Architecture Programming And System Design 8085, 8086, 8051,8096*, PHI Learning Pvt. Ltd.,2010.
3. Hall.D.V., *Microprocessor and Digital System*, McGraw Hill Publishing Company, 2nd edition,2008 .
4. 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 Initilization- 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,Hierarchial,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, 2008.

Reference Books:

1. Schildt Herbert, *The Complete Reference C++*, Tata McGraw Hill, 4th eition, 23rd reprint, 2009.
2. K R Venugopal ,*Mastering C++* ,Tata McGraw-Hill Publication, 2006.

MCA233 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 Methdos, 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 resting, 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 Matrics

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 Matircs-, software measurement-Size Oriented matrics, Function Oriented Matrics, Reconciling LOC and FPMatrics,Object Oriented Matrics,Use case oriented matrics, WebApp project matrics-, 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, Effor 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 International Editions, 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.

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, 6thnd 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 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 behaviour and performance while handling various tasks with respect to different system architecture.

Learning Outcome

Upon completion of the course students will be able to:

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

Unit I. Overview **(10)**

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 (12)

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)**

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 **(13)**

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, 8th edition. 2011, Wiley-India Edition.

Reference Books:

1. William stallings, Operating system – Internals and Design Principles, 7th edition, 2010, Prentice Hall.
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*, 6/E, Thomson, 2011.
4. Dhamdhere, D.M., *Operating Systems: A Concept-based Approach*, 2/E, McGraw Hill, 2006.
5. Dietel D, Operating System, 3rd edition, Pearson Education. 2004.

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 analyzing 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.

Unit I. **(02)**

Accounting

Basic terms -Principles- Concepts - Conventions- IFRS.

Unit II. **(10)**

Double Entry System of accounting

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

Unit III. **(08)**

Final Accounts

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.

Unit IV. **(04)**

Final accounts of Joint Stock Companies

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

Financial Management **(02)**

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

Unit V. **(08)**

Fund Flow Statement

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.

Text 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. Shashi K Gupta, R K Sharma, *Financial Management Theory and Practice* 6th Revised Edition 2010
3. I M Pandey *Management Accounting*, Third revised Edition,2010
4. Lavy and Sarnat, *Principles of Financial Managament*, Prentice Hall.
5. Arnolel, *Financial accounting*, PHI (Paper Back Edition).
6. S N Maheshwari S K Maheshwari, *An Introduction to accountancy* 10 th Edition,2010

Question Paper Pattern

Section A

Answer any Ten questions out of twelve. Each question carries two marks. 10 x 2 = 20

Section B

Answer any four questions out of six. Each question carries ten marks. 4 x 10 = 40

Section C

Answer any two questions out of four. Each question carries 20 marks. 2 x 20 = 40

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 add two digit BCD numbers.
3. Write a program to interchange N one bytes of data.
4. Write a program to check whether the 4th bit of a number is zero or one.
   Display FF if 1 otherwise display 00.
5. Write a program to find the first 10 terms of a Fibonacci sequence
6. Write a program to find sum of first 10 terms of odd and even series.
7. Write a program to check whether a byte belongs to the 2-out-of-5codes. Display FF if it is a 2-out-of- 5 code otherwise00.
   (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)
8. Write a program to perform linear search over a set of N numbers.
   Display FF and its position if found otherwise 00.
9. Write a program to add two 32 - bit binary numbers.
10. Write a program to add two 32 - bit BCD numbers.
11. Write a program to subtract a 16 - bit number from another 16 - bit number.
12. Write a program to subtract a 16 - bit BCD number from another 16 – bit BCD number.
13. Write a program to multiply two 8 - bit number.
14. Write a program to divide a 16 - bit number by an 8 - bit numbers.
15. Write a program to find the largest and smallest of N numbers.
16. Write a program to display a message “HELLO”
17. Write a program to sort the numbers in ascending and in descending and in descending order using bubble sort.
18. Write a program to display a rolling message.
19. Write a program to determine the HCF of two one byte numbers.
20. Write a program to display FF and 00 alternatively with 1.5 sec delay.
21. Write a program to check whether a one byte number is a palindrome or not.
22. Write a program to prepare a look-up table for the squares of one -digit BCD numbers.
23. Write a program to simulate the throw of dice.
24. Write a program to determine the LCM of two one byte numbers.
25. Write a program to simulate a BCD counter to count from 0 to 100.
26. Write a program to simulate a stopwatch with a provision to stop the watch.
27. Write a program to implement block move with the without overlap condition.

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.

MCA271: Seminar-1

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 consultant with the faculty guide. Each Student has to give one hour presentation to their fellow classmates and to a panel of guides.