PREAMBLE

This one year Post Graduate Diploma in Computer Applications (PGDCA) course aims to give adequate expertise to a student to enable him/her to utilise computers for maximum benefit in an Office or Business environment. It will also enable a student to develop programs of his/her own to enhance productivity in such an environment. It will further equip the successful candidate to be able to teach Computer Applications in Higher Secondary Schools. It will provide the necessary skills to make the successful candidates proficient in software development, paving the way for self-employment. The course is oriented more towards Programming and Software than to Hardware.

Eligibility

This course is open to graduates of Arts, Science and Commerce streams. It does not presuppose that the applicant has had prior exposure to Computer Science. The candidate should have a Bachelor Degree of minimum 3 years durations from any recognized university. The candidate must have secured at least 50% marks (45% marks in case of SC/ST)) at the qualifying examinations. However, a good academic record and an aptitude for this subject as indicated by one‟s ability to analyze a problem and identify the sequence of steps required to arrive at a solution would be an essential requirement.

Duration

The Duration of the course is one academic year. Each academic year shall consist of 2(two) semesters. Each semester should have a minimum of 90 working days

Examination

The course is based on the credit system of marking. For theory papers, one credit is assigned for every one hour of class in a week. For practical papers, one credit is assigned for every two hours of class in a week. Evaluation for each course shall be done on the basis of performance in continuous assessment and end-semester examination.The evaluation of the sessional examinations for each paper (other than project work) will be on the basis of student‟s performance in class test/assignments/seminar. For the papers of project work, Sessional examinations will consist of seminar and viva-voce only. There will be three (3) Sessional examinations for each paper. Average of best two performances of the Sessional examinations will be considered for awarding marks in semester examination. End semester practical examination shall be jointly conducted by an external and an internal examiner. If for any reason, the external examiner is not available, a panel of at least two internal examiners shall conduct the practical examination. This panel will be constituted by the Head of the Department. End Semester Examination for project work will be evaluated by a panel of minimum two members consisting of 1(one) external examiner.

The question papers of the external examination will be out of a maximum of 75 marks. However, the marks scored will be converted to grades according to the scheme given below.

Marks Obtained (%)	Grade	Grade Points
75% to 100%	O	5.50-6.00
65% to <75%	A	4.50-5.49
55% to <65%	B	3.50-4.49
45% to <55%	C	2.50-3.49
35% to <45%	D	1.50-2.49
25% to <35%	E	0.50-1.49
0% to <25%	F	0.00-0.49

For awarding grades in any percentage range the following formula shall be applied:
(Base Grade Point of percentage range)+[(the difference between the base percentage and percentage obtained) x (grade range) divided by percentage range]
For example, a candidate securing 63% marks shall be awarded grade as
(3.50) + [(63-55) x 1/10 = 3.5 + 0.8 =4.3 and a student securing 93% marks shall be awarded grade as
(5.50) + [(93-75) x 0.5/25] = 5.86

After the results of a semester are declared, Grade Cards will be issued to each student which will contain the list of papers for that semester and the grades obtained by the student.
The Grade Point Average (GPA) for each semester will be calculated only for those students who have passed all the credit papers of that semester. GPA will be calculated as follows:

$$\sum^{n} C_i * GP_i$$

$$\sum_1 C_i \quad GPA = \underline{\;1\qquad\qquad}$$

where Ci is the number of credits of a paper, GPi is the grade point scored in that paper and n is the total number of papers in a particular semester.
On successful completion of the programme, the Cumulative Grade Point Average (CGPA) will be calculated as follows:

$$CGPA = \frac{\sum C_i * GP_i^N {}_1}{\sum_1^N C_i}$$

where N is the total number of papers for the PGDCA programme.

Requirements for Passing

For all subjects the passing requirement shall be that the candidate secures minimum 40 percent (Grade point 2) in the Internal Assessment as well as in the Semester Examinations. Only the candidates passing in Internal Assessment may be allowed to appear for the Semester Examination.

The marks for internal assessment for each paper are to be given on the basis of

- tests given during the course (3 tests may be given for each subject, and the best two marks may be considered for internal assessment)
- assignments submitted and/or seminars given
- laboratory record book in the papers applicable

A record of this may be maintained in the College.

Practical Record Book : In the papers involving practicals, a standard set of problems have been listed. These and/or others similar to them are to be done as practical work and submitted by the student in a laboratory record book. For each problem, the following sections are to be recorded:

- problem definition
- pseudocode and/or flowchart,
- sample test data
- source code, and
- sample input/output screens.

Classification of Successful Candidates

a) The successful candidates will be given marksheets that will contain the grades they have scored in each paper, as well as the GPA and CGPA.
b) Successful candidates shall be awarded the PGDCA degree according to the following classifications:
(i) First Class: Those who obtain CGPA in the range of 4 and above.
(ii) Second Class: Those who obtain CGPA of 3 to 3.9.
(iii) Pass: Those who obtain CGPA of 2 to 2.9.

OUTLINE OF SYLLABUS

for

POST GRADUATE DIPLOMA IN COMPUTER APPLICATIONS

Semester	Paper		Class	Marks		Credits	Hours
	No	Name	Hours	Internal Assessment	Semester Exam		
1	DCA-101	Digital Logic and Computer Organisation	50	25	75	3	3
	DCA-102	Data Communications and Networking	50	25	75	3	3
	DCA-103	Database Management Systems	50	25	75	3	3
	DCA-104	Programming in C	50	25	75	3	3
	DCA-105	Accounting and Financial Management	50	25	75	3	3
	DCA-106	Practical 1 : Programming in C	50	25	75	2	3
2	DCA-207	Programming for the World Wide Web	50	25	75	3	3
	DCA-208	Programming in VB.Net	50	25	75	3	3
	DCA-209	Software Engineering	50	25	75	3	3
	DCA-210	Practical 2 Programming for the World Wide Web	50	25	75	2	3
	DCA-211	Practical 3 : Programming in VB.Net	50	25	75	2	3
	DCA-212	Project	100	50	50	4	Viva
			630	325	875	34	

DCA-101 : DIGITAL LOGIC AND COMPUTER ORGANISATION

Objective

This paper aims to cover some of the basic concepts of computer organization. It also provides an introductin to digital logic devices and Boolean algebra. The topics included here should enable a student to understand how data is represented and processed within a computer system.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Credits	Marks		
50	3		External	Internal	Total
		3	75	25	100

Units	Topics	Class Hours	Marks
1	Introduction	10	15
2	Digital Devices	10	15
3	Boolean Algebra	14	20
4	Control unit	10	15
5	I/O and Memory units	6	10
Total		50	75

Detailed Syllabus

Unit 1: **10 Hours**

Introduction:
Number systems: Bit, Byte, Nibble, Word, Binary Number, Binary Addition, Binary Subtraction, Binary Multiplication, Binary Division, 1‟s, 2‟s, 9‟s & 10‟s complements. Use of complements to represent negative numbers. Octal, Decimal & Hexadecimal number systems, Conversion of numbers from one number system to another e.g. Conversion of a binary integer to a decimal (base 10) integer and vice versa, Conversion of a binary fraction to a decimal (base 10) fraction and vice versa, BCD, EBCDIC, ASCII & Gray codes.

Unit 2: **10 Hours**

Digital devices: Logic gates (AND, OR, NOT, NAND, NOR, XOR and XNOR), Realization of other logic functions using NAND/NOR gates. Drawing logic diagrams for different types of Boolean expression derived from truth tables; A brief introduction to Combinational and Sequential circuits. Differences between Combinational and Sequential circuits; Arithmetic circuits: Half-adder, Full-adder, Subtractor, Encoders, Decoders, Multiplexer, Demultiplexers; Definition of Flip-flops, Registers. Counters: Asynchronous counters, Synchronous counters.

Unit 3: **14 Hours**

Boolean Algebra: AND, OR, NOT, NAND, NOR, EXOR, EXNOR operations.
The following Theorems of Boolean Algebra
X+0 = X, X.1 = X, X+1 = 1, X.0 = 0, X+X = X, X.X = X, X+X‟ = 1, X.X‟ = 0, X.(Y+Z) = X.Y+X.Z, X+YZ = (X+Y).(X+Z), X+XY = X, X(X+Y) = X, X+X‟Y = (X+Y), X(X‟+Y) = XY, XY+XY‟ = X, (X+Y).(X+Y‟) = X, XY+X‟Z = (X+Z)(X‟+Y), (X+Y)(X‟+Z) = XZ+X‟Y, XY+X‟Z+YZ = XY+X‟Z, (X+Y)(X‟+Z)(Y+Z) = (X+Y)(X‟+Z) & De Morgan‟s theorems.

Simplification of Boolean expression by algebraic method. Complement of a Boolean expression, Sum Of Products and Product Of Sums forms of logic expression, Canonical form of a logic expression, Conversion of Products Of Sums expressions into Sum Of Products form and vice versa. Conversion of Sum of Products expressions into canonical form, Conversion of Products of Sums expressions into Canonical forms. Boolean expression (in Sum Of Products form and Products Of Sums form) from a given Truth Table, Problems on how to obtain Boolean expressions in Product Of Sums or Sum of Products form from a given Truth Table. Implementing logic expression with logic gates (logic expression can either be in Sum Of Product form or Products Of Sums form). Using universal gates to represent any Boolean expression. Convert a logic diagram that uses different types of gates (e.g. AND, OR, NOT gates) into a diagram that uses only one type of universal gate (either ONLY NAND or ONLY NOR gate).

Problems on minimizing Boolean expressions: Minimization using Karnaugh map for two, three and four variables, Realization of the Product of sums expression from Karnaugh map, Realization of the Sum of Products expression from Karnaugh map. Don‟t Care Conditions- problems using Don‟t care conditions, benefit of using Don‟t care conditions.

Unit 4: **10 Hours**

Control unit
Block diagram of a computer.
Brief overview of the use of the following terms- Assembly Language, Machine Language, CPU, ALU, Accumulator, Program Counter, Registers; Instruction Format, Opcode, Operand, Source Operand, Destination Operand, Next Instruction, Stacks. Examples of 0-Addressing, 1-Addressing, 2-Addressing, 3-Addressing. Some Addressing Modes- viz. Immediate, Direct, Indirect, Register and Register-Indirect. Logical shifts- Logical Left, Logical Right, Arithmetic Left, Arithmetic Right, Rotate Left, Rotate Right.

Unit 5: **6 Hours**

I/O unit: One (1) type of I/O operation viz. DMA (Direct Memory Access)- DMA Controller and DMA Transfer.
Memory unit: Stored program concept; Use of Main(RAM, SD RAM, DDRAM, ROM PROM, EPROM, EEPROM), Cache and Secondary memory; L1 & L2 Cache; Numerical problems on Hard Disks.

Instructions to Paper Setter
Questions should be set according to the following scheme. Each question can have parts.

Unit	**Questions**		**Marks**	***Exam Hours***
	To be set	***To be answered***		
1	2	1	15	3
2	2	1	15	
3	3	2	20	
4	2	1	15	
5	2	1	10	

Books For Study:

Text:

1. **Mano, M Morris,** *Digital Logic Design,* **Prentice Hall of India,** NewDelhi-110001
2. **Mano, M Morris,** *Computer System Architecture***, Prentice Hall of India,** NewDelhi-110001

References:

1. **Malvino and Leach**, ***Digital Computer and applications***, Tata McGraw Hill, New Delhi.
2. **Ram, B.**, ***Fundamentals of Microprocessors and Microcomputers***, Dhanpat Raji & Sons, Delhi, 1995.
3. **Hamacher, V. Carl**, ***Computer Organization***, McGraw-Hill, New York, Fourth Edition, 1997.
4. **Bartee, Thomas C**, *Digital Computer Fundamentals*, Mc. Graw Hill, New York.

DCA-102 : DATA COMMUNICATIONS AND NETWORKING

Objective

Data communications and computer networks have become a topic of utmost significance to the present day information based society. This paper intends to provide students an understanding to the basic concepts and mechanisms underlying data communications and networking.

Outline of the Course

Minimum Class Hours	**Exam Time (Hours)**	**Credits**	**Marks**		
50	3		External	Internal	Total
		3	75	25	100

Unit	**Topic**	**Minimum Class Hours**	**Marks**
1	Introduction to Networks	13	20
2	LAN Protocols:	13	20
3	Wide Area Network (WAN)	12	20
4	Network Management and Network k Operating Systems	12	15
TOTAL		**50**	**75**

Detailed Syllabus

Unit 1 **13 Hrs.**

Introduction to networks: Overview of networking – need for networking - hardware and software components; Network communication standards – OSI reference models: Physical layer- Data Link layer– Network Layer – Transport layer –Session layer –Presentation layer – Application Layer; TCP/IP reference model: Application layer – Transport layer – Internet Layer – Network access layer.

Network topologies: Bus topology – Ring topology – star topology – Mesh topologies – Hybrid topologies. LAN components: LAN cables and connectors – cables – Bandwidth – distance – attenuation – coaxial cables – Twisted pair cables – Shielded twisted pair – unshielded twisted pair – Optical fiber cables – Multimode and single mode transmissions; LAN Devices – Repeaters – Hubs – Switches – NICs – Wireless LANs.

Unit 2 **13 Hrs.**

LAN Protocols: Lower layer protocols –ARCnet – Ethernet –Ethernet communication – Fast and Gigabit Ethernets – Token ring –Token ring frame format – Fault management and Tolerance - Fiber Distributed Data Interface; Middle Layer Protocols – Transmission control protocol/internet protocol (TCP/IP) – Internetwork Packet Exchange/ Sequenced Packet Exchange (IPX/SPX)Higher layer Protocols –HTTP – FTP - SMTP –POP – IMAP.

Network Addressing: Introduction – TCP/IP Addressing Scheme – Components of IP address – IP Address classes – Limitations of IP address classes – IP Subnetting – creating subnets in networks – communication across subnets – Subnetting considerations – Subnetting limitations – VLSM – CIDR – IPv6.

Unit 3 **12 Hrs.**

Wide Area Network (WAN) : WAN –WAN Connectivity options – POTS – Leased lines; ISDN – features and working of ISDN; VSAT – Devices – Networks – Architecture – Access Technologies –Microwave – Radio – Infrared; Virtual Private networks (VPNs) – working – Protocols – PPTP - L2TP – IPSec.

WAN Hardware and Protocols: WAN Devices – Bridges – Transparent and source route bridges; Routers – Routing mechanics – Node and router based Routing – Routing Table - Routing Protocols – RIP –OSPF; Gateways; WAN Protocols – PPP – X.25 – Frame Relay – Virtual circuits – Frame Relay Implementation; Asynchronous Transfer Mode (ATM) – Reference Model – Local Area network Emulation.

Unit 4 **12 Hrs.**
Network management: Overview; ISO Network management model – Configuration, Fault, performance, security, and Accounting management. Network Monitoring and Troubleshooting: SNMP – SNMP Manager –SNMP agent – SNMP working; Remote Monitoring (RMON); Ethreal.
Network operating system: Overview – Features of windows 2000 server – Features of Novell Netware.
Internet tools and services: DNS; windows INTERNET Service (WINS) – Working – Name Registration – Name Renewal – Name Release; Dynamic Host Configuration Protocol (DHCP) – working –Phases of leasing IP Address.

Instruction to Paper setter

Unit	**Questions**		**Marks**
	To be set	**To be answered**	
1	2	1	20
2	2	1	20
3	2	1	20
4	2	1	15
Total	**10**	**5**	**75**

Recommended Books

Text

Basics of Networking, NIIT, Prentice Hall of India, New Delhi,2008

Reference

1. **Andrew S. Tenenbaum,** *Computer Networks (Third Ed.)* Prentice Hall of India, New Delhi,2000.
2. **Behrouz A. Forouzan ,** *Data Communications and Networking(Second Ed.)* Tata McGraw Hill.
3. **Sudakshina Kundu,** *Fundamentals of Computer Networks,* Prentice Hall of India**,** New Delhi, 2005

DCA-103 : DATABASE MANAGEMENT SYSTEM

Objective

The objective of this paper is to introduce to the students the fundamental concepts necessary for designing, using and implementing database systems and applications. The paper stresses on database modeling and design, physical file storage techniques and language facilities provided by database management systems. The students are also provided with an overview of some of the emerging database technologies and applications.

Outline of the Course

Minimum Class Hours	Credits	Exam Time (Hours)	Marks		
			Semester Exam	Internal	Total
50	3	3	75	25	100

Unit	Topic	Hours	Marks
1	Introduction and Conceptual Data Modeling	12	20
2	Relational Data Model and Normalization	18	24
3	Structured Query Language	10	15
4	File Organization	10	16
Total		50	75

Detailed Syllabus

Unit 1: Introduction and Conceptual Data Modeling 12 Hours

Introduction: Introduction to databases, characteristics of the database approach, database users and of designers, role of a DBA, advantages using a DBMS, data models, schemas, instances, DBMS architecture (Three-Schema Architecture)
C*onceptual Data Modeling*: Phases of database design, entity type, entity set, attributes, keys, value sets, relationships, relationship types, relationship sets, relationship instances, relationship degree, role names, recursive relationships, constraints on relationship types, attributes of relationship types, weak entity types, ER Diagram, naming conventions and design issues.

Unit 2 : Relational Data Model and Normalization 18 Hours

Relational model concepts: Domain, attribute, tuple, relation, characteristics of relations, relational databases, relational database schemas, relational constraints (Domain constraint, constraints on null), entity integrity, referential integrity, foreign keys. ER to Relational mapping algorithm
Relational Algebra: basic relational algebra operations-SELECT, PROJECT, UNION, INTERSECTION, SET DIFFERENCE, Cartesian PRODUCT, JOIN, Aggregate functions
Normalization: Functional Dependencies, First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Denormalization

Unit 3 : Structured Query Language 10 Hours

SQL: Characteristics of SQL, Data types in SQL, Types of SQL commands
Data Definition Commands: CREATE SCHEMA, CREATE TABLE, DROP TABLE, ALTER TABLE
Single table query commands: SELECT, SELECT with WHERE, SELECT with ORDER BY, SELECT with GROUP BY, SELECT with GROUP BY and HAVING, SQL built-in functions - SUM, MIN, MAX, COUNT, AVG
Multi-table query commands: Retrieval using sub-query, JOIN, EXIST and NOT EXIST
Special operators: IS NULL, IS NOT NULL, BETWEEN..AND, IN, LIKE, ANY, ALL
Data changing commands: INSERT, DELETE, UPDATE

Unit 4 : **File Organization** **10 Hours**

Storage Devices: Introduction to storage hierarchies, hardware descriptions of disk devices, Magnetic Tape Storage Devices, RAID technology (Raid Level 0, Level 1 and Level 2).
File Organization: Organization of file records on disk (record and record types, Fixed- length records, variable-length records, record blocking, spanned and unspanned records, allocating file blocks on disk, file headers, primary methods of file organization -Heap Files, Sorted Files, Hashed Files.
Indexing methods: Primary Indexes, Clustering Indexes, Secondary Indexes, Multilevel Indexes.

Instructions to Paper Setter

Questions should be set according to the following scheme. Each question may have parts.

Unit	**Questions**	
	To be set	**To be answered**
1	3	2
2	3	2
3	2	1
4	2	1

Recommended Books

Text:

1. **R.Elmasri, S.B Navathe**, *Fundamentals of Database Systems*, Addision, Wesley (Third Edition) 2000.

Reference:

1. **A. Silberschatz, H.F Korth, S Sudarshan**, *Database System Concepts***, *Tata***- McGraw Hill, 1997.
2. **Bipin Desai**, *An Introduction to Database Systems*, Galgotia Publications (West Publishing), 1991.
3. **D.M Kroenke**, *Database Processing***:** *Fundamentals, Design and Implementation*, Prentice-Hall of India, (Eighth Edition) 2002.
4. **G.W Hansen, J.V Hansen**, *Database Management and Design***,** Prentice-Hall of India, (2nd Edition) 2001.
5. **Thomas M Connolly, Carolyn E Begg,** *Database Systems, A Practical Approach to Design, Implementation and Management*, Addison Wesley Longman Ltd. 1999.

DCA -104 : PROBLEM SOLVING & PROGRAMMING USING C

Objective

The objective of the course is to introduce the fundamentals of C programming language and develop the skills for solving problems using computers. The course will enable a student to design and program real world problems using C.

Minimum Class Hours	Exam Time (Hours)	Credits	Marks		
			Semester Exam	**Internal**	**Total**
50	3	3	75	25	100

Unit	Topic	Hours	Marks
1	Introduction to Algorithms, C fundamentals	20	25
2	Functions, Storage Class, Arrays and Pointer	15	25
3	Structure ,Union and Files	15	25
Total		50	75

Detailed Syllabus

Unit 1: Introduction to Algorithms, C Fundamentals **20 Hours**
Definition of Program & Algorithm, Pseudocode, Flowchart, Implementation of algorithms
C Fundamentals: The C character set, identifiers and key words, Data types, constants, variables and arrays, declarations, expressions, statements, symbolic constants, Operators and Expressions, Arithmetic operators, unary operators, relational, logical and bitwise operators, assignment operators, library functions.
I/O functions: Preliminaries, getch, getche, getchar, putch, putchar, scanf, printf, gets, puts. Control statements: Preliminaries, while, do.. while, for, if..else, switch, break, continue, goto statements. The C Preprocessor: Macro Expansions.

Unit 2 : Functions, Storage Classes, Arrays and Pointers **15 Hours**
Functions: A brief overview, defining a function, accessing a function, passing arguments to a function, specifying argument data types, function prototypes, recursion, call by value, call by reference. Program Structure: Storage classes, Automatic, Register, External, Static Variables.
Arrays and Pointers : Defining an array, processing an array, passing array to a function, multidimensional arrays, arrays and strings, pointer declarations, passing pointer to a function, pointer and one dimensional arrays, Operation on pointers, pointers and multidimensional arrays.

Unit 3 : Structures, Unions and Files **15 Hours**
Structures and Unions : Defining a structure, processing a structure, user defined data types, structures and arrays, structures and pointers, passing structures to a function, self referential structures, Union.
Data files: File opening modes, Formatted console I/O functions, checking file opening error, closing data files. Command line parameters, low level disk I/O. Enumerations, typedef.

Instructions to Paper Setter
Questions should be set according to the following scheme. Each question may have parts.

Unit	Questions	
	To be set	**To be answered**
1	3	2
2	3	2
3	3	2

Recommended Books

1. **Text: E. Balaguruswamy**, *Programming in ANSI C*, Tata McGraw Hill publication

References:

2. **Byron S. Gottfried**, *Theory and Problems of Programming with C*, Tata McGraw Hill Publication
3. **Yashavant Kanetkar**, *Let us C* , BPB Publication

DCA-105 : ACCOUNTING AND FINANCIAL MANAGEMENT

Objective

The objective of this paper is to make the students familiar with the basic accounting and financial management concepts. This takes into account the knowledge of accounting that a student may require when faced with the task of developing or maintaining any package for any business/financial institutions as well as for non profit organisations.

Outline of the Course

Minimum Class Hours	**Credits**	**Exam Time (Hours)**	**Marks**		
			Internal	**External**	**Tota l**
50	3	3	25	75	100

Unit	**Topics**	**Minimum Class Hours**	**Marks**
1	Fundamentals Principles of Accounting	12	12
2	Final Accounts	12	20
3	Financial Management	12	15
4	Techniques of costing	8	14
5	Budget and Budgetary Control	6	14
Total		50	75

Unit 1 : **12 Hours**

Accounting principles, concepts and conventions, Double entry system of accounting, journal, ledger, trial balance, cash book, single double and triple column cash book, preparation of Bank Reconciliation statement.

Unit 2 : **12 Hours**

Capital and Revenue Expenditure, construction of trading, profit and loss accounts and balance sheet of sole proprietorship concerns with adjustments, manufacturing account, simple problems on final accounts of companies.

Accounts of non trading institutions-preparation of income and expenditure account from receipts and payments account, balance sheet.

Unit 3 : **12 Hours**

Meaning and Role of Financial Management, Ratio Analysis – Meaning, Advantages, limitations, types of ratios and their usefulness, problems on ratio analysis.

Fund Flow Statement- meaning of the term funds, flow and fund flow, preparation of fund flow statement.

Unit 4 : **8 Hours**

Concept of cost, costing and cost accounting, classification of cost, preparation of cost sheet.

Marginal costing – Basic concepts, break-even analysis, construction of break-even chart, problems on marginal costing, application of marginal costing in decision making.

Unit 5 : **6 Hours**

Budget and Budgetary Control – Meaning, importance, types of budget, preparation of flexible budgets and cash budgets.

Instructions For Paper Setter:

The number of questions to be set from each unit and the number of questions that the student has to answer are specified in the table below. Each question may have parts.

Unit	To be set	To be answered
1	2	1
2	2	1
3	2	1
4	2	1
5	2	1
Total	10	5

Books For Study :
Text :

1. Dr. Jawahar Lal, Accounting for Management, Himalaya Publishing House, Mumbai

Reference :

1. C. Mohan Juneja, R.C. Chawla, K.K.Saksena, Double Entry Book Keeping, Kalyani Publishers, Ludhiana.
2. S.P.Jain, K.L.Narang, Cost Accounting, Kalayani Publishers, Ludhiana.
3. Shukla, Grewal, Gupta, Advanced Accounts, Schand & Sons, Delhi

DCA -106 : Practical 1
(PROBLEM SOLVING & PROGRAMMING USING C)

Minimum Class Hours	Exam Time (Hours)	Credits	Marks		
			Semester Exam	Internal	Total
50	3	2	75	25	100

Unit	Topic	Hours	Marks
1	C fundamentals, Functions, Storage Class, Arrays and Pointer	35	50
2	Structure ,Union and Files	15	25
Total		50	75

Detailed Syllabus :

Same as DCA-104

Instructions to Paper Setter

For the examination in practicals, the problems need not be restricted to those given in the syllabus. However, they should be of similar standard. For evaluation of practical examination, the following points may be considered :

* Logic (pseudocode, sourcecode, error trapping)	Weightage	40 percent
* Coding and syntax	Weightage	10 percent
* I/O Design	Weightage	10 percent
* Completion	Weightage	20 percent
* Result	Weightage	20 percent

Questions should be set according to the following scheme.

Unit	Questions	
	To be set	To be answered
1	4	2
2	2	1

Practical Assignments

(Questions may not be restricted to this list)

1. Write a program to display the message “Welcome to the C programming world” on the screen.
2. Write a program to find out the sum of two integer values and display the result on the screen. Input the two values from the keyboard.
3. Write a program to find out the greatest of three numbers.
4. Write a program for **swap**ping the two numbers with / without using another variable.
5. Write a program to find whether the given year is a leap year or not (use % modulus operator)
6. Write a program to find out the real roots of quadratic equation, $Ax^2+Bx+C=0$.
7. Write a program to convert the given temperature in Fahrenheit to Celsius using the following conversion formula, C=(F-32)/1.8.
8. Write a program to generate Fibonacci sequence. (0,1,1,2,3,5,8,13, …)
9. An employee is paid 1.5 times the normal rate for every hour beyond 40 hours worked in a week. Write a program to calculate the weekly wage of an employee.
10. Write a program to check whether a given string is palindrome or not.
11. The total distance traveled by a vehicle in t seconds is given by

    Distance = $ut + (at^2)/2$

Where **u** is the initial velocity (meters per second), **a** is the acceleration (meters per second2). Write a program to evaluate the distance traveled at regular intervals of time, given the values of **u** and **a**. The program should provide the flexibility to the user to select his own time intervals and repeat the calculations for different values of **u** and **a**.

12. For a certain electrical circuit with an inductance L and resistance R, the damped natural frequency is given by
    Frequency $=\sqrt{}$ [(1/LC-R^2/4C^2)]
    It is desired to study the variation of this frequency with C (capacitance). Write a program to calculate the frequency for different value of C starting from **0.1** in steps of **0.01**.
13. Write a program to read the following numbers, round them off to the nearest integers and print out the results in integer form:
    35.7 50.21 -23.73 -46.45
14. Admission to a professional course is subject to the following conditions:
    (a) Marks in mathematics >=60
    (b) Marks in physics >=50
    (c) Marks in chemistry >=40
    (d) Total in all three subjects
    >=200 Or
    Total in mathematics and physics >=150
    Write a program to search of admission of students. The user has to enter the marks from the keyboard of the corresponding subjects.
15. Write a program to calculate the monthly telephone bill according to the following rules:
    (a) Rural subscribers:

Upto 250 calls	Free
251 calls to 450 calls	0.60
451 calls to 500 calls	0.80
501 calls to 1000 calls	1.00
above 1000 calls	1.20

(b) Urban subscribers:

Upto 150 calls	Free
151 calls to 400 calls	0.80
401 calls to 1000 calls	1.00
above 1000 calls	1.20

(c) The rental for urban subscribers depends on the number of calls upto 400 calls the rental will be 200/- and above 400 calls the rental will be 240/-. For rural subscribers the rental is always 200/-.

16. Write a C program to input the Name, City Type (whether Metro or Non-Metro) and Basic Pay of an employee and calculate the salary according to the following rules:
    (a) Dearness allowance (DA)
    (i) Upto Rs. 3500 110% of basic pay
    (ii) Above Rs.3500 90% of the basic pay subject to a minimum of Rs. 3850 (i.e. DA should be at least Rs. 3850.
    (b) House Rent Allowance (HRA) is 15% of the basic pay subject to a maximum of Rs. 800 (i.e. never more than Rs. 800)
    (c) If City is Metro, City Compensatory Allowance (CCA)=800 else if it is Non-Metro, CCA=600.
    (d) Provident Fund (PF) is 12% of the basic pay.
    (**Total Salary**=Basic Pay +DA+HRA+CCA-PF)

    The **output** should be in the following format (Example only)

```
Example Name            ABCDEF
Basic Salary            5000
Dearness Allowance      4500
HRA                     750
CCA: Non-Metro          600
PF                      600
Total Salary            10250
```

17. Write a program to sum the following series:
    a) The first n natural numbers
    b) The first n odd natural numbers
    c) The first n even natural numbers

18. Write a program to sum the series : 2 * 3 – 3 * 5 + 4 * 7 + to n terms
19. Given a number, write a program using while loop to reverse the digits of the number. For example, the number 12345 should be written as 54321. (**Hint:** Use modulus operator to extract the last digit and the integer division by 10 to get the n-1 digit number from the n digit number.)
20. Write a program for sorting the elements of an array by using Selection sort, Bubble sort, Insertion sort.
21. Write a program to generate positive prime numbers.
22. Write a program to display the multiplication table of a given number for a given range.
23. Write a program to find the biggest and smallest number and its position in the given array.
24. Write a program to find add, subtract and multiply two matrices using function.
25. The factorial of an integer m is the product of consecutive integers from 1 to m. That is, Factorial m = m! =m*(m-1)*(m-2)*…*1.
26. Write a program to find the sum of row, column, and diagonals of the given matrix.
27. Write a program to find the largest number of the given matrix using function.
28. Write a program to sort all the elements of a matrix using function.
29. Write a program to input a string and perform the following tasks without using library functions: (a) to find its length, (b) to change it to upper case / lower case (c) to extract the left most n characters, (d) to extract the right most n characters (e) to extract n characters from it starting from position p, (f) to insert another string in it at position p (g) to replace n characters in it starting at position p with a given string
30. Write a program to search a pattern in a given text.
31. Write a program to search a pattern in a given text and replace every occurrence of it with another given string.
32. Write a program to write a given number in words using function.
33. Write a program to display the text in a FILE. (TYPE command in DOS).
34. Write a program to copy the contents of one text file to another using command line arguments.
35. Write a program to merge the two text files into a third text file.
36. Write a program to copy the contents of one text file to any number of given files using command line arguments.
37. Write a program to count the number of characters, lines and words in a text file.
38. Write a program to print every line of a text file containing a given pattern.
39. Write a program to input, sort, and display n names using using array of pointers.
40. Write a menu driven program to create records of students with marks in various subjects and store them in a file. Make provision for viewing all the records, searching a particular record, editing a particular record, deleting a particular record and listing a particular group of records.

DCA-207: PROGRAMMING FOR THE WORLD WIDE WEB

Objective: The objective of this paper is to provide a comprehensive knowledge to the programming tools and skills required to build and maintain Web documents and Websites on the Web. This paper also provides an overview of how the World Wide Web works.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Credits	Marks		
			Semester Exam	Internal	Total
50	3	3	75	25	100

Unit	Topic	Minimum Class Hours	Marks
1	Fundamentals and Introduction to XHTML	10	15
2	Cascading Style Sheets	14	20
3	Basics of JavaScript; JavaScript and XHTML Documents	14	20
4	Introduction to PHP	12	20
TOTAL		**50**	**75**

UNIT 1 — 10 hours

Fundamentals: - A Brief Introduction to the Internet, The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, MIME, HTTP, Security, Overview of XHTML, Tools for creating XHTML Documents, Overview of JavaScript, Overview of PHP.

Introduction to XHTML: - Origins and Evolution of HTML and XHTML, Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, Frames.

UNIT 2 — 14 hours

Cascading Style Sheets: Introduction, Levels of Style Sheets, Style Specification Formats, Selector Forms, Property Value Forms, Font Properties, List Properties, Color, Alignment of Text, The Box Model, Background Images, The <span> and <div> tags, Conflict Resolution.

UNIT 3 — 14 hours

Basics of JavaScript: Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Primitives, Operations and Expressions, Screen Output and Keyboard Input, Control Statements, Arrays, Functions, Pattern Matching Using Regular Expressions.

JavaScript and XHTML Documents: The JavaScript Execution Environment, The Document Object Model, Element Access in JavaScript, Events and Event Handling, Handling Events from Body Elements, Handling Events from Button Elements, Handling Events from Text Box and Password Elements.

UNIT 4 — 12 hours

Introduction to PHP: Origins and Uses of PHP, Overview of PHP, General Syntactic Characteristics, Primitives, Operations and Expressions, Output, Control Statements, Arrays, Functions, Pattern Matching, Form Handling, Database Access with PHP and MySQL.

Instruction to Paper Setter

Unit	Questions	
	To be Set	**To be Answered**
1	2	1
2	2	1
3	2	1
4	2	1
Total	**8**	**4**

Recommended Books

Text:

1. **Robert W. Sebesta, *Programming with World Wide Web*,** Fourth Edition, Pearson Education, New Delhi.

References:

1. **Deitel, H. M; P. J. Deitel, *Internet and World Wide Web: How to Programme***, 2nd Edition, Prentice Hall, India. 2002

DCA-208: PROGRAMMING IN VB.NET

Objective

Visual Basic.NET is the latest version of Visual Basic, the most significant evolutionary change yet in the language. At its heart is the totally new .NET framework, a rich and powerful set of classes that provides support for just about any imaginable area of programming – desktop, Internet, database, and so on. The intent of this course is to teach the language as well as the programming logic, the..NET framework, Database programming, and Web application

Outline of the Course

Minimum Class Hours	**Credits**	**Exam Time (Hours)**	**Marks**		
50	3	3	**External**	**Internal**	**Total**
			75	25	100

Unit	**Topics**	**Minimum Class Hours**	**Marks**
1	Introduction to VB.NET	15	20
2	User-Defined Data Types, Procedures & Exception Handling	10	20
3	Basics of Object-Oriented Programming, MDI Applications & Library Functions	10	20
4	Database & Web Programming	15	15
Total		50	75

Unit 1: Introduction to VB.NET **15 Hours**

Getting Started : Types of VB.NET projects ; IDE Components- IDE Menu, Toolbox window, Solution Explorer, Properties Window, Output window, Command window, TaskList window; Common Properties of Controls – Name, Font, Enabled, Size, Tag, Text, TabStop, TabIndex; Common Events – Click, DoubleClick, Enter, Leave, MouseEnter, MouseLeave; Common Methods- Focus, Clear, Hide, Show, Scale; Building a Console application

Language Fundamentals: Comments, Variables – Declaring variables, Types of variables, Data type identifiers, Strict and Explicit options, Object variables, Converting variable types, widening and narrowing, User-defined data types, Examining variable types, Scope and lifetime of a variable; Constants, Flow control statements – Test Structures – If…Then, If…Then…Else, Select Case, Loop structures – For…Next, Do…Loop, While…End While; Nested Control structures; Exit statement

Basic Windows Controls - Label, TextBox, CheckBox, RadioButton, ListBox, CheckedListBox, ComboBox, GroupBox, TabContol, Timer, StatusBar, ImageList, , PictureBox, HScrollBar, VScrollBar, and, Common Dialog Controls

Unit 2: User-Defined Data Types, Procedures & Exception Handling **10 Hours**

Arrays, Structures and Enumerations: Arrays-Declaring and Initializing arrays, array limits, multi-dimensional arrays, dynamic arrays, arrays of arrays; Creating and using Structures; Creating and using Enumerations

Procedures: Modular coding- Subroutines, Functions, Calling Functions and Subroutines; Arguments-Argument-passing mechanisms, Event-handler arguments, Recursive functions

Exception Handling and Debugging: Types of Errors-Design-time, Runtime and Logic errors; Exceptions and Structured Exception handling; Debugging- Breakpoints, stepping through, local and watch windows

Unit 3: Basics of Object-Oriented Programming, MDI Applications & Library Functions **10 Hours**

Object Oriented Programming: OOP Fundamentals – Class and objects, Creating Classes, Creating Property procedures, Class Methods, Class Constructors, Shared Methods, Shared Variables, Class Access Options MDI Applications: MDI Applications Basics-Building an MDI application, Creating and accessing Child Forms, Ending an MDI application

Library Functions: Char Class – Properties : MaxValue, MinValue; Methods – IsDigit(), IsControl(), IsLetter(), IsLetterOrDigit(), IsLower(), IsUpper(), IsNumber(), IsWhiteSpace(), ToLower(), ToUpper(), ToString(); String Class – Properties: Length,Chars; Methods - Compare(), Concat(), Copy(), EndsWith(), StartsWith(), IndexOf(), LastIndexOf(), Insert(), PadLeft(), PadRight(), Remove(), Replace(), Join(), Split(), SubString(), ToLower(), ToUpper(), Trim(), TrimEnd(), TrimStart(); Math Class – Methods : Abs(), Ceiling(), Floor(), Log(), Max(), Min(), Pow(), Round(), Sign(); DateTime Class – Properties : Date, DayOfWeek, DayOfYear, Hour, Minute, Second, Day, Month, Year, TimeOfDay; Methods : Compare(), DaysInMonth(), IsLeapYear(), Add(), Subtract(), AddDays(), AddHours(), AddMinutes(), AddMonths(), AddSeconds(), AddYears(), Today(), ToLongDateString(), ToLongTimeString(), ToShortDateString(), ToShortTimeString(), ToString();

Unit 4: Database and Web Programming **15 Hours**

ADO.NET: ADO.NET architecture, Creating a Dataset, DataGrid control, Data Binding, BindingContext object, Binding complex controls, DataAdapter object, Command object, DataReader object, ASP.NET : Building a web application, maintaining state, web controls, validation controls

Instructions For Paper Setter:

The question paper will be set according to the following scheme. Each question may have parts.

Theory			**Marks**
Unit	**To be set**	**To be answered**	
1	2	1	20
2	2	1	20
3	2	1	20
4	2	1	15
Total		**4**	**75**

Recommended Books
Text:

1. **Petroutsos, E.,** *Mastering Visual Basic.NET*, New Delhi: BPB Publications, 2004

Reference:

1. **Anne Prince ,** *Murach's Beginning Visual Basic.NET,* Tech Publications PTE Ltd, Singapore, 2002
2. **Steven Holzner** , *Visual Basic .NET Programming, Black Book*, Dreamtech, New Delhi, 2009
3. **Jeffrey R Shapiro**, *Complete Reference, Visual Basic.NET,*
4. **Bill Evjen et al**, *Visual Basic .NET Programming Bible,* IDG Books India (P) Ltd., New Delhi, 2002
5. **Pooja Bembey et al**, *Microsoft Visual Basic .NET Professional Projects,* Prentice Hall of India Pvt. Ltd., New Delhi, 2002
6. **Michael Halvorson**, *Microsoft Visual Basic .NET Step by Step*, Prentice Hall of India Pvt. Ltd., New Delhi, 2002

DCA-209 : SOFTWARE ENGINEERING

Objective

Software Engineering is a fast developing field. We can view Software Engineering as the engineering approach to developing software. The objective of this paper is to provide a broad understanding of system development concepts. It provides the students with a sense of confidence to develop new systems.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Credits	Marks		
			Semester Exam	Internal	Total
50	3	3	75	25	100

Unit	Topic	Minimum Class Hours	Marks
1	Introduction To Software Engineering Software life cycle models	8	15
2	Software project management, requirements and design	18	25
3	Function Oriented Software Design and user interface Design	14	20
4	Testing, software reliability and maintenance	10	15
TOTAL		50	75

UNIT 1 **8 hours**

Introduction: - Evolution of an art to an engineering disciple, Solution to the software crisis,
Computer systems engineering.
Software Life cycle models: - Importance of a life cycle model, waterfall model (feasibility study, requirement analysis and specification, design, coding and unit testing integration and system testing, maintenance), prototyping model, evolutionary model, spiral model, Comparison of different life cycle models.

UNIT 2 **18 hours**

Software project management: Responsibilities of a software project manager, project planning, project estimation techniques, COCOMO, Scheduling (work breakdown, Activity Networks and critical Path Method, Gantt Charts, PERT Charts, project Monitoring and control), Organization and team structures (Organization structure, team structure), Risk management (Risk identification, risk assessment, Risk Containment), Software configuration Management (Necessity of software Configuration Management, Configuration Management Activities, Source code control system and RCS.
Requirement Analysis and specification: Requirement gathering and analysis, Software requirements specification (Content of the SRS document, Functional Requirements, how to identify the functional requirements, how to document the functional requirements, traceabilty, characteristics of a good SRS document, techniques for representing complex logic).
Software Design: introduction of a good software design, cohesion and coupling (classification of cohesiveness and coupling), Software designs approaches (function-oriented design, Object-oriented Design).

UNIT 3 **14 hours**

Function Oriented Software Design: Overview of SA/SD methodology, Structured Analysis, data Flow Diagrams (DFDs)(primitive symbols used for constructing DFDs, important concepts associated with designing DFDs, developing the DFD Model of a system, Shortcomings of the DFD Model), Extending DFD technique to real-time systems, Structured Design (flow chart vs. structure chart, transformation of a DFD model into a structure chart), Detailed Design, Design Review.

User Interface Design: characteristics of a good user interface, basic concepts (user guidance and online help, mode-based vs. Modeless Interface, Graphical User Interface (GUI) vs. Text-based User Interface), Types of user interfaces (command language-based Interface, Menu-based Interface, direct manipulation Interface), Component-Based GUI Development (Window system, Types of widgets, an overview of X-window/Motif, X-Architecture, Visual programming), User interface methodology (Design, a GUI design methodology, Task and object modeling, selecting a metaphor, interaction design and rough layout, user interface inspection).

UNIT 4 **10 hours**

Coding and Testing: coding standards and guidelines, code review (code walkthroughs, code inspection, clean room testing, software documentation), Testing (testing, verification vs. validation, design of test cases), Testing in the large, Testing in the small, unit testing, black-box testing, White-box testing, debugging, program analysis tools, integration testing, system testing.

Software Reliability: Software reliability, statistical testing, software quality, and software quality management.

Software Maintenance: Characteristics of software maintenance (types of software maintenance, characteristics of software evolution, special problems associated with software maintenance), software reverse engineering, software maintenance process models, estimation of maintenance cost.

Instruction to Paper Setter

Unit	Questions	
	To be Set	To be Answered
1	2	1
2	2	1
3	2	1
4	2	1
Total	**8**	**4**

Recommended Books

Text:

2. **Rajib Mall, *Fundamentals of Software Engineering*,** Prentice Hall of India Private Limited, New Delhi.

References:

3. **Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli**, ***Fundamentals Of Software Engineering***, Second Edition, Prentice Hall of India Private Limited, New Delhi, 2002.
4. **Richard E Fairley**, ***Software Engineering Concepts***, Tata McGraw Hill Publishing Company Limited, New Delhi, 1997.

DCA-210 : Practical 2 (PROGRAMMING FOR THE WORLD WIDE WEB)

Minimum Class Hours	**Exam Time** (hours)	**Credits**	**Marks**		
			External	**Internal**	**Total**
50	3	2	75	25	100

Unit	**Topic**	**Minimum Class Hours**	**Marks**
1	Fundamentals and Introduction to XHTML	10	75
2	Cascading Style Sheets		
3	Basics of JavaScript; JavaScript and XHTML Documents	30	
4	Introduction to PHP	10	
TOTAL		**50**	**75**

Detailed Syllabus :

Same as DCA-207

Instructions to Paper Setter

For the examination in practicals, the problems need not be restricted to those given in the syllabus. However, they should be of similar standard. For evaluation of practical examination, the following points may be considered :

*	Logic (pseudocode, sourcecode, error trapping)	Weightage	40 percent
*	Coding and syntax	Weightage	10 percent
*	I/O Design	Weightage	10 percent
*	Completion	Weightage	20 percent
*	Result	Weightage	20 percent

Questions should be set according to the following scheme. There can be parts for each question.

Unit	**Questions**		**Time (Hours)**	**Marks**
	To be set	**To be answered**		
1	2	1	3	25
2				
3	3	2		50
4				
Total	**5**	**3**	**3**	**75**

Practical Assignments : Sample Programs

(Questions may not be restricted to this list)

List of Practical Exercises for Programming World Wide Web:

1. Create an XHTML document for yourself, including your name, address, e-mail address, phone number, date of birth and age. If you are a student, you must include the course you have undertaken and describing a little bit about the course. If you are employed, you must include your employer, your employer‟s address and your job title. This document must use several headings and <em>, <strong>,<hr/>,<p> and <br/> tags.
2. Create an XHTML document to describe an unordered list of a typical grocery shopping list you write.
3. Create an XHTML document to describe an unordered list of at least four states. Each element of the list must have a nested list of at least three cities in the state.
4. Create an XHTML document to describe an ordered list of your favorite top ten movies.

5. Modify the list of Exercise 4 to add nested, unordered lists of at least two actors and/or actresses in your favorite movies.
6. Create an XTHML document to describe an ordered list with the following contents: the highest level should be the names of your two parents, with your mother first. Under each parent, you must have a nested, ordered list with the brothers and sisters of your parents, in order with the eldest first. Each of the nested lists must have nested lists that list the children of your uncles and aunts (your cousins)-under the proper parents of course. Regardless of how many aunts, uncles and cousins you actually have, there must be at least three list items in each sublist below each of your parents and below each of your aunts and uncles.
7. Create an XHTML document that defines a table with columns for state, state bird/animal, state flower and state food. There must be at least five states as rows in the table.
8. Create an XHTML document that has a form with the following controls:
    a. A text box to collect user"s name.
    b. Four checkboxes, one each for the following items:
        i. Four 100-watt light bulbs for Rs70.
        ii. Eight 100-watt light bulbs for Rs140.
        iii. Four 100-watt long-life light bulbs for Rs90.
        iv. Eight 100-watt long-life light bulbs for rs210
    c. A collection of three radio buttons that are labeled as follows:
        i. Visa
        ii. Mastercard
        iii. Maestro.
9. Create an XHTML document that has six short paragraphs of text that describe various aspects of the state in which you live. You must define three paragraph styles p1,p2 and p3. The p1 style must use left and right margins of 20 pixels, background color of pink and foreground color of blue. The p2 style must use left and right margins of 30 pixels, background color of black and foreground color of yellow. The p3 style must use a text indent of 1 cm, background color of green and foreground color of white. The first and fourth paragraph must use p1, the second and fifth must use p2 and the third and sixth must use p3.
10. Create an XHTML document that describes nested ordered lists of cars. The outer list must have three entries: compact, midsize and sports. Inside each of these three lists, there must be two sublists of body styles. The compact and midsize car sublists are two-door and four-door, the sports car sublists are coupe and convertible. Each body-style sublist must have at least three entries, each of which is the make and model of a particular car that fits the category. The outer list must use uppercase Roman numerals, the middle lists must use uppercase letters and the inner list must use Arabic numerals. The background color for the compact car list must be pink; for the midsize car list it must be blue; for the sports car list, it must be red. All of the styles must be in a document style sheet.
11. Create an XHTML document that contains an unordered list of at least five popular books. The bullet for each book must be small image of the book"s cover.
12. Write a Javascript code to display a table of the numbers from 5 to 15 and their squares and cubes through an XHTML document. [Hint: Use document.write to display output in a tabular form, using the assistance of table HTML tags]. Use for loop or do loop.
13. Write a Javascript code to display the first 50 Fibonacci Numbers through an XHTML document. [Hint: Use document.write to display output in a tabular form, using the assistance of table HTML tags]. Use for loop or do loop.
14. Write a Javascript code to display a list of Armstrong numbers between 100 and 1000 through an XHTML document. [Hint: Use document.write to display output in a tabular form, using the assistance of table HTML tags]. Use for loop or do loop.
15. Write a Javascript code to display a table of Palindrome numbers between 100 and 500 through an XHTML document. [Hint: Use document.write to display output in a tabular form, using the assistance of table HTML tags]. Use for loop or do loop.

16. Write a Javascript code to generate a list of numbers between 100 and 1000 where the result of the current number is the sum of the previous four numbers in the series. Example initial four numbers are 0,1,1,2. The next number in the series should be 4.
17. Write a Javascript code to display a list of odd numbers from 1 to 100.
18. Write a Javascript code to display all prime numbers between 1 and 100.

19. A company wants to transmit data over the telephone, but it is concerned that its phones may be tapped. All of its data are transmitted as four-digit integers. It has asked you to write a program that will encrypt its data so that the data may be transmitted more securely. Your script should specify the input as 4561 and encrypt it as follows: Replace each digit by *(the sum of that digit plus 7) modulus 10*. Then swap the first digit with the third, and swap the second digit with the fourth. Then output XHTML text that displays the encrypted integer.
20. Use a single-subscripted array to solve the following problem: A company pays its salespeople on a commission basis. The salespeople receive $200 per week plus 9% of their gross sales for that week. For example, a salesperson who grosses $5000 in sales in a week receives $200 plus 9% of $5000, or a total of $650. Write a script (using an array of counters) that obtains the gross sales for each employee through an XHTML form and determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson‟s salary is truncated to an integer amount):
    a) $200-$299
    b) $300-$399
    c) $400-$499
    d) $500-$599
    e) $600-$699
    f) $700-$799
    g) $800-$899
    h) $900-$999
    i) $1000 and over

21. Use a single-subscripted array to solve the following problem: Read in 20 numbers, each of which is between 10 and 100, inclusive. As each number is read, print it only if it is not a duplicate of a number that has already been read.
22. Use the program pieces developed in parts (a) and (b) to write a function **displayDigits** that receives an integer between **1** and **99999** and prints it as a series of digits, each pair of which is separated by two spaces. For example, the integer **4562** should be printed as **4 5 6 2**.
23. Write a Javascript code that creates customer‟s bills for a carpet company when the following information is given as input:

    The length and width of the carpet in feet.
    The carpet price per square foot.
    The percentage of discount for each customer.
    The labor cost is fixed at Rs.10 per square foot and the tax rate is 8.5%. The bill should display the amounts under various heads like carpet cost, labor cost, discount, tax and total amount.
    Total amount=carpet cost+labor cost – discount + tax.

    Use arrays for inputting the number of customers whose bills are to be calculated.
24. The document must have a paragraph of text that describes your home. Choose at least three different phrases (three to six words) of this paragraph and make them change font, font style, color and font size when the mouse cursor is placed over them. Each of the different phrases must change to different fonts, font styles, colors and font sizes.

DCA-211 : Practical 3 - PROGRAMMING IN VB.NET

Outline of the Course

Minimum Class Hours	Credits	Exam Time Hours	Marks		
			External	**Internal**	**Total**
50	2	3	75	25	100

Unit	Topics	Minimum Class Hours	Marks
1	Starting with Visual Basic Programming	15	25
2	Using Procedures and Functions	15	25
3	Creating Menus, using Rich Text Box and Common Dialog Box Control, MsFlexGrid and Database Programming	20	25
Total		50	75

Objective:

The main objective of this paper is to give a strong practical foundation in Visual programming including Crystal Report, Menu Design & DataBase Management. It helps to develop powerful applications with less programming effort. It will be the stepping stone to the next generation programming.

(Note: Practical problems need not be restricted to these lists)

Unit I

1. Design a form and place a TextBox in it. Call it (assign its name property) txtInput. Place a Command Button and call it cmdExtract. Assign the caption property of the Command button as "Extract". Write a program to extract each digit or letter of a number, word or sentence that is entered in txtInput and display them in a second Text Box called txtOutput one at a time on the click of a button.

2. A frmEmployee contains a TextBox (txtNumber) to enter number of employee records to be entered and two Command Buttons (cmdOK with the caption Ok and cmdClose with caption Close). As soon as a single digit number is entered, appropriate number of controls must be available in the form for entering Name, Address, Salary for the given number of employees. Write the code in appropriate Event to accomplish these.

3. Write a program to verify whether a given date is a valid date or not. Do not use library functions.

4. Design a form with suitable controls to input a single digit number and write appropriate event handlers to check if the number is automorphic or not. A number is called automorphic if the last digit of the square of the number is same as the number itself.(e.g., 6)

5. Design a form with suitable controls and write appropriate event handlers to list out all the Armstrong numbers within a given range of numbers „m‟ to „n‟. A number is called an Armstrong number if the number is equal to the sum of the cubes of the digits of the number.

6. Design a form with suitable controls and write appropriate event handlers to generate an Ordinary Calculator Program (Using Label, CommandButton). The calculator should support the facilities such as Addition, Subtraction, Multiplication, Division, Storing in Memory, Clearing Memory and Adding to Memory etc. The display of the calculator should support up to 10 digits including decimal point. Your application should use control arrays

7. In the color code that is used in resistors, the different colors have values as follows: Black=0, Brown=1, Red=2, Orange=3, Yellow=4, Green=5, Blue=6, Violet=7, Gray=8 and White=9. The value of the resistor is indicated by drawing three colored bands round it. The first two bands indicate the first two digits in the numerical value of the resistance, while the third band is the decimal multiplier, i.e., it gives the number of zeros after the two digits. For example, if the bands have colors, Green-Blue-Orange, successively, then the numerical value is 56000. Design a form with suitable controls and write appropriate event handlers to accept the colors from the user and print the equivalent numerical.

8. Using functions, write a program to calculate the simple interest accrued on a given principal using the formula SI = (Principal x Rate x Time)/100. The user input and the output thereof must be on different forms. The input form must have a textbox where the principal will be entered by the user, a vertical scroll bar for the rate of interest, and a listbox from where the user can select the time (in years.) On clicking a button, the function must calculate the SI taking values from the textbox, scrollbar, and listbox, and the result shown in the second form. Provision must also be kept for adding and removing items to and from the listbox. The items in the listbox appear as: 1 year; 2 years; 3 years etc.…. up to 10 years.
9. Write a program to calculate and display the factorial of a given number, using a recursive function.
10. Design a form with suitable controls and write appropriate event handlers to convert an input decimal number to a number with a user defined base (1 to 9), and vice versa.
11. Develop an application providing the facilities for a stopwatch, a timer, and a daily alarm at a preset time, as desired by the user. The selection of the option should be through radio buttons.
12. Load a picture on an appropriate control such that the position of the picture randomly changes within the form with time.
13. Develop an application where all possible colours can displayed in a picture box using the three primary colours red, green, blue, whose values are selected from three scrollbars.
14. Load a picture on an appropriate control such that the position of the picture randomly changes within the form with time.
15. Write a program to sort the elements of an array in descending order using bubble sort, selection sort and insertion sort.
16. Write a program to search for an element in an array using binary search and linear search.

Unit II

17. Design a form with suitable controls and write appropriate event handlers to take in a string and determine whether the given string is palindrome or not.
18. Design a form with suitable controls and write appropriate event handlers to generate the calendar of a given month. The user must enter the month and the year. Assume that 1st January 1900 was a Monday.
19. Write a program to search for a particular word or pattern in a text and to display the position of the match. The match should also be selected.
20. Write a program to convert a string to proper case.
21. Write a program to input a string and perform the following tasks without using library functions: (a) to find its length, (b) to change it to upper case / lower case (c) to extract the left most n characters, (d) to extract the right most n characters (e) to extract n characters from it starting from position p, (f) to insert another string in it at position p (g) to replace n characters in it starting at position p with a given string
22. Write a program to analyze a line of text i.e to count no. of words, digits, letters, special characters, vowels, consonants, no. of times a particular word appears.
23. Write a line of text and place it centered on a form. Ensure that the text remains centered even if the form is resized manually or otherwise.
24. A line of text E.g.“ Over to Delhi for the second day‟s play.” is entered by the user. Write a program to print the shortest and longest word so contained in the sentence.
25. Create a class rectangle. The class has attributes length and width, each which defaults to 1. It has methods that calculate the perimeter and the area of the rectangle. It has set and get methods for both length and width. The set method should verify that length and width are each floating point numbers larger than 0.0 and less than 20.0. Create a class to test this Rectangle class.
26. Create a more sophisticated Rectangle class which stores only the Cartesian coordinates of the four corners of the rectangle. The constructor calls a set method that accepts four sets of coordinates and verifies that each of these are in the first quadrant with no single x or y coordinate more than 20.0. The set method also verifies that the supplied coordinates specify a rectangle. Provide methods to calculate

the length, width, perimeter and area. The length must be the larger of the two dimensions. Include a Boolean method isSquare which determines if the rectangle is a square. Write a program to test this class.

27. Create a class savingAccount. Use static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingBalance by annualInterestRate divided by 12- this interest must be added to savingBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test this class.

28. Create a class called complex for performing arithmetic with complex numbers of the form – real + imaginary * i, where i = √-1

    Write a program to test your class. Use floating point variables to represent the private date of the class. Provide a constructor that enables an object of this class to be initialized when created. Provide a no-argument constructor with default values in case no initializers are provided. Provide public methods that perform the following operations.

    i. Add two complex numbers. (a+bi) + (c+di)=(a+c) + (b +d) i
    ii. Subtract two complex numbers. (a+bi) - (c+di)=(a-c) + (b -d) i
    iii. Multiple complex numbers (a+bi) * (c+di)=(a*c –b*d) + (a*d+b*c) i
    iv. Print complex numbers.

29. An educational institution wishes to maintain a database of its employees. A staff member has a code and name. a staff member can be either a teacher, with a subject, or a typist, with typing speed, or an officer, with a grade. A typist can be a regular typist with a basic salary or a casual employee with daily wage. Write a program to define all these classes and also define methods to store and retrieve values for these classes.

30. Assume that a bank maintains two kinds of accounts for its customers, one called saving account and the other current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks –

    - Accept deposit from a customer and update his balance.
    - Display the balance
    - Compute and deposit interest
    - Permit withdrawal and update balance
    - Check for the minimum balance, impose penalty, if necessary, and update balance.

Unit III

31. The following information is to be maintained regarding the users of electricity: Name, code and units consumed. Write a program that will take the name and units consumed, and hence generate a bill. For the first 20 units cost is 30p/unit, for the next 20 units, 40p/unit, for the rest, 50p/unit. Make provisions for reading, editing and deleting data. Make provisions to keep the rates alterable.

32. A publishing company maintains records with the following information: Name of Author, Author Code, Name of Book, Book ID and Year of Publication. Make provisions to add, edit and delete records. Every time a new Author Name is added, the code must be generated, so also with the name of book and book ID.

33. Refer to the above question to design a Crystal Report to display the details of the books for a given Author and given Year of Publication. Design a VB.NET form and write appropriate code to invoke the report.

34. Create a Database in a Database Server with two tables –Biodata and Marks. The table Biodata contains the fields Name, RollNo(N, 5)(RollNo is unique), Gender(C,1), State(C,15), District(C,15), Place(C,15), Class(C,3),Dob(Date), Caste(C,10) and the table Marks contains the fields RollNo(N,5), Physics(N,2), Chemistry(N,2), Maths(N,2). Develop ASP page(s) to Add, Edit and Delete records from the table. Provision should also be made to display all the records of a given class, along with each ones‟ average mark, in a tabular format (the class can be selected from a listbox).

35. Develop a web page that will calculate the monthly installment for a loan amount, given the rate of interest and its term.

36. Develop a web page with a counter that displays the number of visits to the site.

37. Develop a program to let user place order for ice cream over the net. This should allow selection of one or more flavours (vanilla, strawberry, etc.) and then select the item (cone, double cone, cups, etc.). The order summary should be displayed on the page once the user clicks on the *Order* button.

12. Develop a web site for a commercial organisation, where the order for goods can be placed. There should be possibility for adding new items or removing items from the shopping cart.

38. Develop a web site for registering the details of alumni for an educational institution. Make provisions for listing out the entries belonging to a particular batch.

Distribution of marks for practical

For the examination in practicals, the problems need not be restricted to those given in the syllabus. However, they should be of similar standard. For evaluation of practical examination, the following points may be considered :

* Logic (pseudocode, sourcecode, error trapping)	Weightage	40 percent
* Coding and syntax	Weightage	10 percent
* I/O Design	Weightage	10 percent
* Completion	Weightage	20 percent
* Result	Weightage	20 percent

Instructions For Paper Setter:

Questions should be set according to the following scheme.

Questions			Marks
Unit	To be set	To be answered	
1	2	1	25
2	2	1	25
3	2	1	25

DCA-212 : PROJECT

Objective

The project aims to consolidate the concepts and practices that were imparted during the course and to serve as a record of competence. It should enable a student to apply concretely in a small package the concepts gained from System Analysis and Design.

Guidelines

The project will be carried out over a duration of 2 months, involving about 120 hours. Every student should do a project individually and not in a group, under the guidance of at least one of the faculty members and/or expert/professional from an organisation outside of the College. The Project Report should be submitted covering the aspects specified under Project Profile below. The Project Report should have a certificate from the College stating it to be a bonafide work of the student that it has not been submitted for any other examination.

Project Profile

Model 1

1. The topic for the project can be any subsystem of a system software or tool or any scientific or a fairly complex algorithmic situation.

2. The aim of this type is to highlight the abilities of algorithmic formulation, program and data flow representation, modular programming, optimised code preparation, systematic documentation and other associated aspects of software engineering.

3. The assessment would be through the Project Report and Viva. The former should portray the following:

   - Programming style, structured design, minimum coupling and high cohesion
   - Good commenting and annotating of the code and flow of representation, such that meaningful code, with good readability and ease of maintenance, results.
   - Design specifications, depicting the method adopted and giving a simple data dictionary, for each data, to cover name, type and validity aspects.
   - Test case samples, enough in number, to adequately cover the possible chances of common errors
   - User manual

Model 2

1. This model can be of a typical business application. The aim of this type is to highlight the stages involved in a typical business oriented project development, though on a miniature scale and simulated environment. The appropriate use of DBMS/RDBMS towards any business application, along with adequate level system analysis and structured design and development of specific tools/products would be the underlying activity, in preparing this project.

2. The emphasis should be on selecting a system/subsystem which shows the DBMS and System Analysis aspects to a greater degree. Any small and simple business system may be selected, although candidates are advised to use their knowledge and creativity, to select typical and intelligent applications, rather than run-of-the-mill themes, such as simple Pay roll calculation or Issue-Return portion of an inventory scheme. The Evaluation stage would give due weightage for theme selection, problem analysis, fact finding techniques and initial design, which is as close to real-life business situations as possible.

3. The code for the project can be generated out of 4 GL Interface, like Form Designer and Report Generator, Application Generator/Program Code Generators. The documentation need not contain the

code generated by these applications, but only that written by the candidate. In the place of the code generated by these applications, the screens they generate can be included.

4. The assessment would be through the Project Report and Viva. The former should portray the following:

    - Requirements leading to the project, those which were the result of System Analysis
    - The design aspect of DBMS oriented documentation which describes the structure and organisation of the database well annotated source code, supplemental documentation, which can serve as a Data Analysis and Data Flow description
    - A simple Data Dictionary of the elements, which form the structure. The number of tables/files which make up the DBMS, should not be less than three.
    - Details about I/O Screens and facilities, for on-screen querying, printed oriented Reports and built in house-keeping routines, which help disk management and file integrity, are to be included to a limited extent
    - User Manual

Viva-Voce

The viva-voce will be conducted by an external examiner appointed by the University and an internal examiner from the College. Other members of the faculty and students may be present. It will be of a duration of about 15 to 20 minutes. The logic, analysis and design aspects relevant to the project mentioned under assessment would be the main subject matter for the viva. However the general proficiency of the candidate in the selected software platform should also be tested.

Documentation

System Overview will have the following sections:

1. Application area : eg Production/Inventory / Finance / Marketing / Human Resource Management / Library / Training/Service Sector/System Level Programming, etc
2. System/Subsystem : eg Invoicing & Accounts Receivables, Purchase & Accounts Payables, Budget & Accounts with variance analysis, Production & Sales Monitoring, Material Requirement Planning, Hotel Management/Hospital Management/Specific Software utility
3. End User(s) : Finance Controller, Marketing Manager, Production Manager, Personnel Department, School or College or Hospital administrate staff, etc
4. Analysis and description of the system, and the specific gains from computerisation
5. Description of the various modules
6. Performance criteria for the proposed system : eg volume of transactions (data handling), control aspects, timeliness, archival
7. Need for review : deficiencies and future enhancements

Data Dictionary

1. This should give a catalogue of the data elements used in the system/subsystem developed.
2. The following are the details required. Write NA where NOT applicable
    - Data Name
    - Aliases, if any
    - Length (size)
    - Type (Numeric, alpha, binary, etc)
    - Validity criterion (Minimum, maximum, etc)
    - Default value, if any
    - Whether related to other data items
    - Where used in the program:
      reference to data structure/file/procedures/modules

User Manual :

It may include chapters like the ones suggested below:

1. Installation
2. Hardware requirements
3. System requirements
4. Installation procedure, including security aspects like password, backups, controls, etc
5. Menu choices and their actions - screen formats
6. Error messages
7. Output
8. A Sample test case

Distribution Of Marks :

Project Report	: 25
Viva	: 25
Internal Assessment	: 50