SCHEME OF EXAMINATION

&

DETAILED SYLLABUS

(w. e. f. Academic Year 2010-2011)

For

MASTER OF COMPUTER APPLICATIONS (MCA) DEGREE

Second Semester

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY
KASHMERE GATE, DELHI - 110403
SCHEME OF EXAMINATIONS

Master of Computer Applications

SECOND SEMESTER EXAMINATION

Paper ID	Paper Code	Paper	L	T/P	Credit
044102	MCA 102	Data and File Structures	3	1	4
044104	MCA 104	Object Oriented Programming in C++	3	1	4
044106	MCA 106	Operating Systems	3	1	4
044108	MCA 108	Database Management Systems	3	1	4
044110	MCA 110	Software Engineering	3	1	4
Practical					
044152	MCA 152	Data and File Structures Lab	0	2	1
044154	MCA 154	Object Oriented Programming in C++ Lab	0	4	2
044156	MCA 156	Database Management Systems Lab	0	2	1
044158	MCA 158	Software Engineering Lab.	0	2	1
NUES					
044162	MCA 162	General Proficiency – II* (It is suggested to have Personality Development and Communication Skills – II Course)	0	2	1
		Total	**15**	**17**	**26**

*** Non-University Examination System (NUES)**

Code No.: MCA 102 **L T C**
Paper: Data and File Structures **3 1 4**

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVE: *In this course student will become familiar with Algorithm analysis: Trees, Graphs, searching and sorting and files.*

PRE-REQUISITES:

- C Programming
- Basic Concepts in Data Structure
- Prelims of Trees and Graphs Functionality of Group Theory

UNIT – I

Fundamentals of algorithm analysis Big 'O' notations, Time and space complexity of algorithms, linked lists: singly and doubly linked lists, stacks, queues, double stack, multistacks and multiqueues, deques, polynomial arithmetic, infix, postfix and prefix arithmetic expression conversion and evaluations. **[No. of Hrs: 08]**

UNIT – II

Trees: Binary trees: Definition, Binary Search Tree basic operations, Tree Traversals (recursive and stack based non-recursive), Heaps and priority queues, Threaded binary tree, AVL Trees B-Tree: need, properties, creation, uses. B+ tree, B* tree. **[No. of Hrs: 10]**

UNIT – III

Graphs: Representation (Matrix and Linked), Traversals, Connected components, Spanning trees, Shortest path and Transitive closure, Topological sort, Activity network, Critical path, Path enumeration. Dijkstra's Algorithm, Floyd Warshall's Algorithm, Coloring of Graphs, Spanning Tree, Minimum Spanning Tree Algorithms (Kruskal's Algorithm, Prim's Algorithm)

Searching & Sorting: Binary search, Hash function, Hash table, Search tree. Internal sort: Radixsort, Insertion sort, Selection sort, Shell sort, Quick sort, Merge sort, Heap sort.

[No. of Hrs: 16]

UNIT – IV

Files: Sequential file organization, creating updating retrieving from sequential files advantages and disadvantages of sequential file organization. Data representation and denisity, parity and error control techniques, devices and channels, double buffering and block buffering, handling sequential files in C language, seeking, positioning, reading and writing binary files in C. External Sorting and merging files k way and polyphase merge **[No. of Hrs: 08]**

TEXT BOOKS:

1. E. Horowitz and S. Sahani, "Fundamentals of Data Structures in C", 2nd Edition, Universities Press, 2008.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison-Wesley, 1997.

REFERENCES:

1. Schaum's Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.
2. Y. Langsam et. al., "Data Structures using C and C++", PHI, 1999.
3. N. Dale and S.C. Lilly, D.C. Heath and Co., "Data Structures", 1995.
4. R. S. Salaria, Khanna, "Data Structure & Algorithms", Book Publishing Co. (P) Ltd., 2002.
5. Richard F. Gilberg and Behrouz A. Forouzan, "Data Structure A Pseudocode Approach with C", Cengage Learning, 2nd Ed., 2005.
6. Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

Code No.: MCA 104 **L T C**
Paper: Object Oriented Programming in C++ **3 1 4**

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVES: *After covering the core C++ in about 25 lectures the course shall aim to aquaint the students about advanced features of the language the following features are as suggested guideline for the teacher.*

- *Copy construtor, Deep and shallow coping, assignment operator and destrutors, when the programmer must implement these*
- *Static and late binding. Run time and comile time polymorphism, virtual functions and VTABLE*
- *Implementing ADT with C++ classes. Stacks Queues and Linked Lists as cases*
- *Implementing Trees and Graph and all comparison based sorting algorithms*
- *Function objects and call backs*
- *Templates and Generics Stack Queses should be implemented in the practicals*
- *Extensive coverage of all the three components of STL namely containers, iterators and algorithms throgh suitable pratical caseletts*
- *Final case study could be an application making extensive handling files.streams classes*

PRE-REQUISITES:

- Data Structure Concept
- Real Programming Experience with C Language
- UNIT-III of MCA-102 should be finished before start of Unit-IV of this paper

UNIT – I

OOP Paradigm: Comparison of Programming paradigms, Characteristics of Object-Oriented Programming Languages, Object-based programming languages **C++**: Brief History of C++, Structure of a C++ program, Difference between C and C++ - cin, cout, new, delete operators, ANSI/ISO Standard C++, Comments, Working with Variables and const Qualifiers. Enumeration, Arrays and Pointer.

Implementing oops concepts in c++ Objects, Classes, Encapsulation, Data Abstraction, Inheritance, Polymorphism, Dynamic Binding, Message Passing, Default Parameter Value, Using Reference variables with Functions. **[No. of Hrs: 09]**

UNIT – II

Abstract data types, Class Component, Object & Class, Constructors Default and Copy Constructor, Assignment operator deep and shallow coping, Access modifiers – private, public and protected. Implementing Class Functions within Class declaration or outside the Class declaration. instantiation of objects, Scope resolution operator, Working with Friend Functions, Using Static Class members. Understanding Compile Time Polymorphism function overloading Rules of Operator Overloading (Unary and Binary) as member function/friend function, Implementation of operator overloading of Arithmetic Operators, Overloading Output/Input, Prefix/ Postfix Increment and decrement Operators, Overloading comparison operators,

Assignment, subscript and function call Operator , concepts of namespaces. **[No. of Hrs. 10]**

UNIT – III

Inheritance: Inheritance, Types of Inheritance, Abstract Classes, Ambiguity resolution using scope resolution operator and Virtual base class, Aggregation, composition vs classification hierarchies, Overriding inheritance methods, Constructors and Destructor in derived classes. Multiple Inheritance.

Polymorphism: Polymorphism, Type of Polymorphism – compile time and runtime, Understanding Dynamic polymorphism: Pointer to objects, Virtual Functions (concept of VTABLE) , pure virtual functions, Abstract Class.

Advanced Input/Output, Exception Handling and Manipulating strings, Using istream / ostream member functions, Using Manipulators, Creating Manipulator Functions, Understanding Implementation of Files, Writing and Reading Objects. Understanding of working and implementation of Exception Handling. **[No. of Hrs: 11]**

UNIT – IV

Generic Programming: and mastering STL Understanding Generic Functions with implementation of searching sorting algorithm. Overloading of Function Templates. Understanding Class Templates using Implementation of Generic stack, linked lists: singly and doubly linked lists, Binary Search Tree basic operations. Understanding Inheritance with Generic Class.

Standard Template Library:– Understanding Components of Standard Template Library, Working of Containers, Algorithms, Iteraters and Other STL Elements. Implementation of Sequence and Associative containers for different Algorithms using their Iterator. Understanding of Algorithms Requiring Operations on the element using function objects. Implementing graph algorithm dfs, bfs, minimum spanning tree , dijkastra etc using STL **[No. of Hrs: 12]**

TEXT BOOKS:

1. A. R. Venugopal, Rajkumar, and T. Ravishanker "Mastering C++", TMH, 1997.
2. S. B. Lippman and J. Lajoie, "C++ Primer", 3rd Edition, Addison Wesley, 2000.
3. Bruce Eckel, "Thinking in C++", President, Mindview Inc., Prentice Hall, 2nd Ed.

REFERENCE:

1. D . Parasons, "Object Oriented Programming with C++", BPB Publication.
2. Bjarne Stroustrup , "The C++ Programming Language", Addison Welsley, 3rd Ed.
3. Steven C. Lawlor, "The Art of Programming Computer Science with C++", Vikas Publication.
4. Schildt Herbert, "C++: The Complete Reference", Tata McGraw Hill, 4th Ed., 1999.
5. Behrouz A. Forouan, Richrad F. Gilberg, Computer Science - A Structural Approach Using C++", Cengage Learning, 2004.
6. Nell Dale, "C++ Plus Data Structure", Jones and Bartlett, 4th Ed., 2010.
7. Nell Dale, Chips Weens, "Programming and Problem Solving with C++", Jones and Bartlett , 5th Ed., 2010.

Code No.: MCA 106 **L T C**
Paper: Operating Systems **3 1 4**

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVE: *The objectives of this course are to:*

- *Help students become familiar with the fundamental concepts of operating system.*
- *Help students become competent in recognizing operating systems features and issues.*
- *Provide students with sufficient understanding of operating system design and how it impacts application systems design and performance.*

Upon successful completion of this course, the student shall be able to:

- *Exhibit familiarity with the fundamental concepts of operating systems.*
- *Exhibit competence in recognizing operating systems features and issues.*
- *Apply a mature understanding of operating system design and how it impacts application systems design and performance.*

PRE-REQUISTES:

- Basics of Computer System Architecture
- C/C++ Programming Skills

UNIT – I

Operating System: Introduction, Role, Types of OS; Batch Systems, multi programming, time–sharing parallel, distributed and real-time systems, Operating system structure, Operating system components and services, System calls.

Processes: Process Concept, Process Scheduling, Operation on Processes, Cooperating Processes, Threads.

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Algorithm Evaluation. **[No. of Hrs.:10]**

UNIT – II

Interprocess Communication and Synchronization: Background, The Critical-Section Problem, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Critical Regions, Monitors, Message Passing.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Combined Approach to Deadlock Handling.

Memory Management: Background, Logical vs. Physical Address space, swapping, Contiguous allocation, Paging, Segmentation, Segmentation with Paging.

Virtual Memory: Demand Paging, Page Replacement, Page-replacement Algorithms, Performance of Demand Paging, Allocation of Frames, Thrashing, Other Considerations, Demand Segmentation. **[No. of Hrs: 11]**

UNIT – III

Device Management: Techniques for Device Management, Dedicated Devices, Shared Devices, Virtual Devices; Device Characteristics-Hardware Consideration, Channels and Control Units,

Independent Device Operation, Buffering, Multiple Paths, Block Multiplexing, Device Allocation Consideration

Secondary-Storage Structure: Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk Reliability. **[No. of Hrs.: 12]**

UNIT – IV

File-System Interface: File Concept, Access Methods, Directory Structure.

File-System Implementation: Introduction, File-System Structure, Basic File System, Allocation Methods, Free-Space Management, Directory Implementation.

Security : The Security problem, Goals of protection, Access matrix, Authentication, Program threats, System threats, Intrusion detection , Cryptography.

Case Study: Linux Operating System and Windows XP. **[No. of Hrs.: 10]**

TEXT BOOKS:

1. Silbersachatz and Galvin, "Operating System Concepts", John Wiley, 8th Ed., 2009.
2. Milan Kovic., "Operating Systems", Tata McGraw Hill, 2001
3. Deitel, Deitel and Choffnes, "Operating Systems", Pearson ,3rd Edition

REFERENCES:

1. Tannenbaum, "Operating Systems", PHI, 4th Ed., 2000.
2. Madnick E. and Donovan J., "Operating Systems", Tata McGraw Hill, 2001.
3. Flynn McHoes, "Operating System", Cengage Learning, 2006.
4. Pbitra Pal Choudhury, "Operating System Principles and Design", PHI, 2009.
5. Sibsankar Halder and Alex A. Aravind, "Operating System", Pearson, 2009.
6. William Stallings, "Operating Systems Internals & Design Principles", Pearson Education, 6th Ed., 2009.

Code No. MCA 108 **L T C**
Paper: Data Base Management Systems **3 1 4**

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVES: *The purpose of this course is to enable the students know about the fundamental concepts necessary for designing, using and implementing database systems and applications. It also covers advanced techniques and technologies.*

PRE-REQUISITE:

- Elementary Maths (Sets, Relations)
- Basic Data Structure Concepts

UNIT - I

Basic concepts: database & database users, characteristics of the database, database systems, concepts and architecture, date models, schemas & instances, DBMS architecture & data independence, database languages & interfaces, data modeling using the entity-relationship approach. Overview of hierarchical, Network & Relational Data Base Management Systems**.**

[No. of Hrs. 9]

UNIT - II

Relational model, languages & systems: relational data model & relational algebra: relational model concepts, relational model constraints, relational algebra, SQL- a relational database language: date definition in SQL, view and queries in SQL, specifying constraints and indexes in sql. **[No. of Hrs. 12]**

UNIT - III

Oracle Architecture, Logical Data Structures Physical Data Structure, Instances, Table Spaces, Types of Tablespaces, Internal Memory Structure, Background Processes, Data Types, Roles & Privileges, Stored Procedures, User Defined Functions, Cursors, Error Handling, Triggers.

[No. of Hrs. 10]

UNIT - IV

Relational data base design: function dependencies & normalization for relational databases: functional dependencies, normal forms based on primary keys, (1NF, 2NF, 3NF & BCNF), lossless join and dependency preserving decomposition. Concurrency control & recovery techniques: concurrency control techniques, locking techniques, time stamp ordering, granularity of data items, recovery techniques: recovery concepts, database backup and recovery from catastrophic failures. Concepts of object oriented database management systems, Distributed Data Base Management Systems. **[No. of Hrs. 11]**

TEXT BOOKS:

1. Elmsari and Navathe, "Fundamentals of Database Systems", Pearson Education, 5th Ed., 2006.
2. Korth, Silberschatz, "Fundamentals of Database System Concepts", TMH, 6th Ed., 2010.
3. Desai, B., "An Introduction to Database Concepts", Galgotia.
4. Sham Tickoo and Sunil Raina, "Oracle 11g with PL/SQL Approach", Pearson, 2010.

REFERENCES:

1. Date C. J., "An Introduction to Database Systems", Narosa Publishing, 7th Ed., 2005.
2. S. K. Singh, "Database Systems: Concept, Design, and Applications", Pearson's Education, 1st Ed., 2008.
3. Kiffer, "Database Systems: An Application oriented Approach", Pearson.
4. Ullman J. D., "Principals of database systems", Galgotia .
5. Shio Kumar Singh, "Databases Systems Concepts, Design and Applications," Pearson, 2006.

Code No. MCA 110 **L T C**
Paper: Software Engineering **3 1 4**

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVE: *Course is intended to help students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain. The concept covered in syllabus are*

- *The software development process.*
- *Software requirements and specifications.*
- *Software design techniques.*
- *Techniques for developing large software systems.*
- *CASE tools and software development environments.*
- *Software testing, documentation and maintenance.*

PRE-REQUISITE:

- Program Development
- Basic Concepts of Data Management

UNIT-I

Introduction: Software Crisis, Software Processes & Characteristics, Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models.

Software Requirements analysis & specifications: Requirement engineering, requirement elicitation techniques like FAST, QFD & Use case approach, requirements analysis using DFD, Data dictionaries & ER Diagrams, Requirements documentation, Nature of SRS, Characteristics & organization of SRS, Requirement Management, IEEE Std. for SRS. **[No. of Hrs.: 10]**

UNIT-II

Software Project Planning: Size Estimation like lines of Code & Function Count, Cost Estimation Models, COCOMO, Putnam resource allocation model, Validating Software Estimates, Risk Management.

Software Design: Cohesion & Coupling, Classification of Cohesiveness & Coupling, Function Oriented Design, Object Oriented Design. **[No. of Hrs.: 12]**

UNIT-III

Software Metrics: Software measurements: What & Why, Token Count, Halstead Software Science Measures, Data Structure Metrics, Information Flow Metrics.

Software Reliability: Importance, Hardware Reliability & Software Reliability, Failure and Faults, Reliability Models- Basic Model, Logarithmic Poisson Model, Software Quality Models, CMM & ISO 9001. **[No. of Hrs.: 10]**

UNIT-IV

Software Testing: Testing process, Design of test cases, Introduction to functional testing & Structural testing, Unit Testing, Integration and System Testing, Debugging, Alpha & Beta Testing.

Software Maintenance: Management of Maintenance, Maintenance Process, Maintenance Models, Regression Testing, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation. **[No. of Hrs.: 10]**

TEXT BOOKS:

1. K. K. Aggarwal and Yogesh Singh, "Software Engineering", New Age International, 3rd Ed., 2005.
2. R. S. Pressman, "Software Engineering – A Practitioner's Approach", McGraw Hill Int. , 5th Ed., 2001.
3. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa, 3rd Ed., 2005.

REFERENCES:

1. Stephen R. Schach, "Classical & Object Oriented Software Engineering", IRWIN, 1996.
2. James Peter, W. Pedrycz, "Software Engineering: An Engineering Approach", John Wiley & Sons.
3. I. Sommerville, "Software Engineering", Addison Wesley,8th Ed., 2009.
4. Frank Tsui and Orlando Karan, "Essentials of Software Engineering", Joes and Bartlett, 2nd Ed., 2010.
5. Kassem A. Saleh, "Software Engineering", Cengage Learning, 2009.
6. Rajib Mall, "Fundamrntal of Software Engineering", PHI, 3rd Ed., 2009.
7. Carlo Ghizzi , Mehdi Jazayeri and Dino Mandrioli, " Fundamental of Software Engineering", PHI, 2nd Ed.,2003.
8. Carol L. Hoover, Mel Rosso-Llopart and Gil Taran, "Evaluating Project Decision Case Studies in Software Engineering", Pearson, 2010.

There will be following Practical:

1. Data and File Structure Lab **MCA 112**
2. Object Oriented Programming in C++ Lab **MCA 114**
3. Database Management System Lab **MCA 116**
4. Software Engineering Lab **MCA 118**

Code No. : MCA 162
Paper: General Proficiency – II*

It is suggested to have a fundamental course on Personality Development and Communication Skills – II in this semester.

This paper is under Non University Examination system its detail content will be decided by the respective Institute, under approval of the coordination committee based on the requirement of individual institution.

***Non University Examination Scheme (NUES)**

There will not be any external examination of the university. The performance of the candidates should continuously be evaluated by an internal committee. The committee may conduct viva-voce at the end for the award of the marks.