UNIVERSITY OF DELHI

MASTER OF COMPUTER APPLICATIONS

(MCA)

Semester II

Semester II					
	Number of core courses	4			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Practical	Tutorial	**Total**
MCAC201	Data Structures	4	1	0	**5**
MCAC202	Database Systems	4	1	0	**5**
MCAC203	Software Engineering	4	1	0	**5**
MCAC204	Data Communication and Computer Networks	4	1	0	**5**
	Total credits in core course	20			
	Number of elective courses	0			
	Total credits in elective courses	0			
	Number of open electives	1			
	Credits in each open elective	Theory	Practical	Tutorial	**Total**
	Open Elective 2	3	1/0	0/1	**4**
	Total credits in open elective	4			
	Total credits in Semester II	**24**			

List of Open Elective Courses for Semester II		
Course Code	**Course Title**	**Th-P-T**
MCAO201	Java Programming	3-1-0
XXXXX	Open Elective from other department	3-X-X

6. Elizabeth Tebeaux and Sam Dragga, **The Essentials of Technical Communication,** Oxford University Press, 2015.

7. Caroline Tagg, **Exploring Digital Communication: Language in Action,** Routledge, New York, 2015.

SEMESTER – II

MCAC201: DATA STRUCTURES [4-1-0]

Course Outcomes:

On completion of this course, the student will be able to:

CO1: develop programs using basic data structures: sets, lists, stacks, queues, trees, graphs and advanced data structures like balanced trees and skip lists.

CO2: understand the behaviour and application of advanced data structures like Tries, Prefix- and Suffix-trees.

CO3: identify best suited data structure for the problem at hand.

CO4: identify the programming constructs to optimize the performance of the data structure in different scenarios.

Syllabus:

Unit-I Basic data Structures: Primitive Data Types, Abstract Data Types, Arrays - Static and Dynamic, 2D Arrays, Linked Lists - Single, Doubly-linked, Circular; Stacks and Queues using arrays and linked lists; operations, their analysis; Applications to searching & sorting.

Unit-II Trees: Binary Tree, Binary Search Tree, Height Balanced Trees: AVL/RB Tree, 2-3Trees, B and B+ Trees, Splay Trees, Heaps, Priority Queues, Mergeable heaps, Tries, Prefix and Suffix Trees, Skip Lists; operations, their analysis, applications to searching.

Unit-III Sets: Sets, Multisets, Maps, Hash Tables, Dictionaries.

Unit-IV Graphs: Representation of Graphs, Searching in Graphs – BFS and its applications, DFS and its applications.

Readings:

1. Goodrich, M., Tamassia, R. and Mount D, **Data Structures and Algorithms in C++/Java**, 2nd Edition, 2016, Wiley.

2. Elliot B. Koffman, Paul A.T. Wolfgang, **Objects, Abstraction, Data Structures and Design Using C++/Java**, 1st Edition, 2005, Wiley Global Education.

3. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, **Introduction to Algorithms**, 3rd Edition, 2010, Prentice-Hall of India Learning Pvt. Ltd.

MCAC202: DATABASE SYSTEMS [4-1-0]

Course Outcomes:

On completion of this course, the student will be able to:

CO1: understand basic database concepts, including the structure and operation of the relational data model.

CO2: apply logical database design principles, including E-R/EE-R diagrams, conversion of ER diagrams to relations.

CO3: understand the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.

CO4: construct simple and moderately advanced database queries using Structured Query Language (SQL).

CO5: understand the concept of a database transaction including concurrency control, backup and recovery, and data object locking.

CO6: design and implement database projects.

Syllabus:

Unit-I Basic Concepts: Data modeling for a database, abstraction and data integration, three level architecture of a DBMS.

Unit-II Database Design: Entity Relationship model, Extended Entity Relationship model.

Unit-III Relational Model & Relational Data Manipulations: Relation, conversion of ER diagrams to relations, integrity constraints, relational algebra, relational domain & tuple calculus.

Unit-IV Structured Query Language: DDL, DML, Views, Embedded SQL.

Unit-V Relational Database Design Concepts: Functional dependencies, determining keys, normalization-, lossless join and dependency preserving decomposition.

Unit-VI Transaction Management: ACID properties, Concurrency Control in databases, transaction recovery.

Unit-VII Introduction to NoSQL databases, XML databases.

Readings:

1. A. Silberschatz, H. Korth and S. Sudarshan**, Database System Concepts,** 6th Edition**,** McGraw Hill, 2014.

2. R. Elmasri and S. B. Navathe, **Fundamentals of Database Systems**, 7th Edition, Pearson, 2016.

3. R. Ramakrishnan and J. Gehrke, **Database Management Systems,** 3rd Edition, McGraw Hill, 2014.

4. Philip Lewis, Arthur Berstein and Michael Kifer, **Databases and Transaction Processing-An application oriented approach**, Prentice Hall, 2003.

MCAC203: SOFTWARE ENGINEERING [4-1-0]

Course Outcomes:

On completion of the course, the student is expected to:

CO1: demonstrate an understanding of software engineering layered technology and software process models that provide a basis for the software development lifecycle.

CO2: apply agile development methods for developing software.

CO3: describe software/system requirements and understand the processes involved in the discovery and documentation of these requirements.

CO4: practice system modeling techniques and object-oriented design for software development.

CO5: test software using verification and validation, static analysis, reviews, inspections, and audits.

CO6: appreciate software project management that includes project planning, project estimation techniques, risk management, quality management, and configuration management.

CO7: work as an individual and/or in team to develop and deliver quality software.

Syllabus:

Unit-I Software Engineering: The software crisis, principles of software engineering, programming-in-the-small vs. programming-in-the-large.

Unit-II Software process: The software lifecycle, the waterfall model and variations, risk-driven approaches, introduction to evolutionary and prototyping approaches, agile process models, system classifications.

Unit-III Project management: Relationship to lifecycle, project planning, project control, project organization, risk management, cost models, configuration management, version control, quality assurance, metrics.

Unit-IV Software requirements: Requirements analysis, functional and non-functional requirements elicitation, analysis tools, requirements definition, requirements specification, static and dynamic specifications, requirements review.

Unit-V Software design: Design for reuse, design for change, design notations, design evaluation and validation.

Unit-VI Implementation and Maintenance: Programming standards and procedures, modularity, data abstraction, static analysis, unit testing, integration testing, regression testing, verification and validation, tools for testing, fault tolerance, The maintenance problem, the nature of maintenance, planning for maintenance**.**

Readings:

1. R.S. Pressman, **Software Engineering: A Practitioner's Approach** (7th ed.), McGraw-Hill, 2010.

2. I. Sommerville, **Software Engineering** (10th ed.), Pearson Education, 2015.

3. R. Mall, **Fundamentals of Software Engineering** (4th ed.), Prentice-Hall of India, 2014.

4. K.K. Aggarwal and Y. Singh, **Software Engineering** (3rd ed.), New Age International Publishers, 2008.

5. P. Jalote, **An Integrated Approach to Software Engineering** (3rd ed.), Narosa Publishing House, 2005.

6. N.S. Godbole, **Software Quality Assurance: Principles and Practice for Students**, Alpha Science International Limited, 2004.

MCAC204: DATA COMMUNICATION AND COMPUTER NETWORKS [4-1-0]

Course Outcomes:

On completion of this course, the student will be able to:

CO1: apply data communication techniques in real-life experiments like telemetry and also develop some basic skills to modify the existing ones to better suit them in different situations.

CO2: develop expertise and skills to apply services of various types of computer networks in various technical and professional fields.

CO3: reduce the overheads of different Reference models and optimize their performances.

CO4: develop some basic skills to apply, modify and develop new protocols in different layers of existing protocol stacks to suit customized requirements.

CO5: use various network applications to avail network services efficiently and also develop basic skills to design new applications to open new services.

Syllabus:

Unit-I Data Communication Techniques: Theoretical basis of data communication, analog and digital signals, time domain and frequency domain analysis, frequency spectrum and bandwidth, asynchronous and synchronous transmission, data encoding and modulation techniques, baseband and broadband transmission, pulse code modulation, baud rate and bitrate of a channel, multiplexing-FDM & TDM, transmission medium, transmission errors – error detection techniques.

Unit-II Network Classification and Network services: Local Area Networks, Metropolitan Area Networks, Wide Area Network, wireless networks, internetworking and Internet, business and home applications, mobile user services.

Unit-III Network Architecture and Reference Models: Layered network architectures, protocol hierarchies, interface and services, ISO-OSI reference model, TCP/IP reference model, Internet protocol stack.

Unit-IV Datalink Layer Functions and Protocols: Framing, flow-control, error recovery protocols, Data link layer of internet-PPP protocol.

Unit-V Medium Access Sublayer: CSMA/CD protocol and Ethernet, hubs and switches, fast Ethernet, gigabit Ethernet, CSMA/CA protocol and WLAN.

Unit-VI Network and transport layers functions and protocols: Network switching mechanisms-Circuit switching, packet switching, routing and congestion control, TCP/IP protocol architecture.

Unit-VII Network Applications: File transfer protocol, electronic mail, World Wide Web.

Readings:

1. A S Tanenbaum, **Computer Networks**, 5th Edition , Pearson Education India, 2013

2. Behrouz A Forouzan, **Data Communications and Networking**, 5th Edition, McGraw Hill Education, 2017.

MCAO201: JAVA PROGRAMMING [3-1-0]

Course Outcomes:

On completion of this course, the student will be able to:

CO1: understand the object-oriented concepts – Classes, Objects, Inheritance, Polymorphism– for problem solving.

CO2: handle program exceptions.

CO3: design, implement, document, test, and debug a Java application consisting of multiple classes.

CO4 : handle input/output through files.

CO5 : create Java applications with graphical user interface (GUI).

Syllabus:

Unit-I Introductory Concepts: program, identifiers, variables, constants, primitive data types, expressions, control statements, structured data types, arrays, functions.

Unit-II Object Oriented Concepts: Abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, overloading, Abstract classes, Interfaces and Packages.

Unit-III File Handling: Byte Stream, Character Stream, File I/O Basics, File Operations, Serialization.

Unit-IV Exception handling: Throw and Exception, Throw, try and catch Blocks, Multiple Catch Blocks, Finally Clause, Throwable Class, Types of Exceptions, java.lang Exceptions, Built-In Exceptions.

Unit-V GUI Design: GUI based I/O, Input and Message Dialog boxes, Swing components, Displaying text and images in window.

Readings:

1. James Gosling, Bill Joy, Guy L. Steele Jr, Gilad Bracha, Alex Buckley, **The Java Language Specification, Java SE 7 Edition**, Addison-Wesley, 2013**.**

2. Cay S. Horstmann, Core Java - Vol. I – Fundamentals, 10th Edition, Pearson, 2017**.**

3. Deitel & Deitel, **Java-How to Program** (9th ed.), Pearson Education, 2012.

4. Richard Johnson, An Introduction to Java Programming and Object-Oriented Application Development, **Thomson Learning,** 2006.

5. Herbert Schildt, Java: The Complete Reference, 10th Edition, McGraw-Hill Education, 2018.

SEMESTER – III

MCAC301: DESIGN AND ANALYSIS OF ALGORITHMS [4-1-0]

Course Outcomes:

On completion of this course, the student will be able to:

CO1: describe the following algorithm design techniques: iteration , divide and conquer, dynamic programming, greedy approach algorithms.

CO2: analyse the strengths and weaknesses of each technique.

CO3: identify and apply technique(s) suitable for simple applications.

CO4: demonstrate correctness of algorithms and analyse their time complexity theoretically as well as practically.

CO5: model simple problems as graphs and solve them using Graph Algorithms.

CO6: analyze algorithms in the probabilistic framework.

CO7: appreciate that certain problems are too hard to admit fast solutions

Syllabus:

Unit-I Review of Growth of Functions

Unit-II Iterative Algorithms: Searching and Sorting Techniques - Linear search, Binary search, insertion sort – time complexity and proof of correctness.

Unit-III Divide and Conquer: Recurrence Relation, Master's Theorem, Recursion Trees; Binary Search, Merge sort and Quick sort – time complexity and proof of correctness.

Unit-IV Lower bounding techniques: Decision Trees.

Unit-V Linear Sorting: Count Sort, Radix Sort, Bucket Sort.

Unit-VI More on Divide and Conquer: Integer Multiplication, Convolution and Fast-Fourier Transform.