M.Sc (Computer Science)

Scheme of Studies for Master Degree Computer Science Program

Abdul Wali Khan University Timergara Campus

Introduction:

The Department of Computer Science, Abdul Wali Khan University Timrgara Campus offers M.Sc in Computer Science. M.Sc. (Computer Science) is a four-semester course spread over the period of two years. This program is designed to offer in depth knowledge of recent technologies in the field of IT. Students are provided extensive laboratory training on the course content and the current requirements of industries and R and D. In the final semester every student has to undertake a project, which is either an industrial project or research project. In industrial project the students develop software that automates some activity of the organization. In research project the students select an area of their interest and then start in depth study in the selected area. At the end of the semester the students are required to submit a research report.

The course provides exposure to the students to the technologies in-vogue and trains them to take up projects relevant to the industrial needs, the R& D activities and self –employment opportunities. The student after passing the M.Sc course has many opportunities of employment, self-employment and higher studies.

Objectives of the Courses:

The course is designed with a view to cater to the present day requirements in Industries, R and D fields, higher studies and Self-employment. Moreover the course structure intends to inculcate strong practical skills, so that the student can take up independent projects which will help them to be a successful software engineer as well as entrepreneur. The students passed out from the course will serve as quality human resource to take up the state of art research work of the Department.

Title of the Course:

M.Sc. (Computer Science)

Eligibility of the Course:

B.Sc (Computer Science) with 45 % Marks

M.Sc Semester Wise Courses Break Up

First Year

Semester	Code	Title	Crd. Hours
1st	MSC111	Object Oriented Programming	3
	MSC112	Introduction to Information and Communication Technologies	3
	MSC113	Digital logic and Design	3
	MSC114	Data Communication	3
	MSC115	Technical and Business Writing	3
	MSC116	Web Fundaments	3
			Total=18
2nd	MSC121	Data Structures	3
	MSC122	Computer Networks	3
	MSC123	Web Engineering	3
	MSC124	Software Engineering-I	3
	MSC125	Database –I	3
	MSC126	Computer Architecture	3
			Total=18

Second Year

3rd	MSC231	Operating System Concepts	3
	MSC232	Analysis of Algorithms	3
	MSC233	SW-II	3
	MSC234	Theory of Automata and Formal languages	3
	MSC235	Artificial Intelligence	3
	MSC236	Data Base-II	3
			Total=18
4th	MSC241	Visual Programming	3
	MSC242	Computer Graphics	3
	MSC243	Computer Organization and Assembly Language	3
	MSC244	Compiler Construction	3
	MSC245	Master Project	6
			Total= 18

1st SEMESTER

Course Name: Object Oriented Programming **Course Code: MSC111**	
Course Structure: Lectures: 2, Labs: 1	**Credit Hours:** 3
Prerequisites: Programming Fundamentals	
Objectives: The course aims to focus on object-oriented concepts, analysis and software development.	
Course Outline: Evolution of Objecgt Oriented (OO) programming, OO concepts and principles, problem solving in OO paradigm, OO programme design process, classes, methods, objects and encapsulation; constructors and destructors, operator and function overloading, virtual functions, derived classes, inheritance and polymorphism. I/O and file processing, exception handling	
Reference Material: 1. C++ How to Program, 6/E **(Harvey & Paul) Deitel & Deitel** ISBN-10: 0136152503 ISBN-13: 9780136152507 Publisher: Prentice Hall 2. Java How to Program, 7/E **(Harvey & Paul) Deitel & Deitel** ISBN-10: 0132222205 ISBN-13: 9780132222204 Publisher: Prentice Hall	

Course Name: Introduction to Information and Communication Technologies	
Course Structure: Lectures: 2 Labs: 1	**Credit Hours:** 3
Pre-requisite: None	**Semester: 1 Course Code: MSC112**
Course Description: This is an introductory course on Information and Communication Technologies. Topics include ICT terminologies, hardware and software components, the internet and world wide web, and ICT based applications. After completing this course, a student will be able to: • Understand different terms associated with ICT • Identify various components of a computer system • Identify the various categories of software and their usage • Define the basic terms associated with communications and networking • Understand different terms associated with the Internet and World Wide Web. • Use various web tools including Web Browsers, E-mail clients and search utilities. • Use text processing, spreadsheets and presentation tools • Understand the enabling/pervasive features of ICT	
Course Contents: : Basic Definitions & Concepts : Hardware: Computer Systems & Components : Storage Devices , Number Systems	

: Software: Operating Systems, Programming and Application Software
: Introduction to Programming, Databases and Information Systems
: Networks
: Data Communication
: The Internet, Browsers and Search Engines
: The Internet: Email, Collaborative Computing and Social Networking
: The Internet: E-Commerce
: IT Security and other issues
: Project Week
: Review Week

Text Books/Reference Books:

Introduction to Computers by Peter Norton, 6th International Edition (McGraw HILL)
Using Information Technology: A Practical Introduction to Computer & Communications by Williams Sawyer, 6th Edition (McGraw HILL)
Computers, Communications & information: A user's introduction by Sarah E. Hutchinson, Stacey C. Swayer
Fundamentals of Information Technology by Alexis Leon, Mathewsleon Leon press

Course Name: Digital Logic Design	**Course Code: MSC113**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Introduction to Information and Communication Technologies	

Objectives: This course introduces the concept of digital logic, gates and the digital circuits. Further, it focuses on the design and analysis combinational and sequential circuits. It also serves to familiarize the student with the logic design of basic computer hardware components.

Course Outline: Overview of Binary Numbers, Boolean Algebra, switching algebra, and logic gates, Karnaugh Map and Quin-McCluskey methods, simplification of Boolean functions, Combinational Design; two level NAND/NOR implementation, Tabular Minimization, Combinational Logic Design: adders, subtracters, code converters, parity checkers, multilevel NAND/NOR/XOR circuits, MSI Components, design and use of encoders, decoders, multiplexers, BCD adders, and comparators, Latches and flip-flops, Synchronous sequential circuit design and analysis, Registers, synchronous and asynchronous counters, and memories, Control Logic Design. Modern trends in memory design

Reference Material:

1. Digital Design, 4/E **M. Morris Mano** & **Michael D. Ciletti** ISBN-10: 0131989243 ISBN-13: 9780131989245 Publisher: Prentice Hall

2. *Digital Fundamentals* / 9E By Thomas L. Floyd Published by Floyd Publisher, 2007

Course Name: Data Communication Course Code: **MSC114**	
Course Structure: 3	**Credit Hours:** 3
Pre-requisite:	
Objectives: To provide knowledge of Data Communication and different mechanisms of communication	
Course Outlines: Introduction, Data and Network, Layers, OSI Model, Introduction to Signals, Transmission Media, Digital Transmission, PAM, PCM, ASK, FSK, PSK, QAM, Data Communication Techniques and technologies, Modulation, Multiplexing, Types of errors, Data Communication Protocols, Current technologies being used for data communication.	
Reference Material: 1) Behrouz A. Forouzan, Data Communication and Networking, 3rd Edition. 2) William Stalling, Business Data Communication.	

Course Name: Technical and Business Writing **Course Code: MSC115**	
Course Structure: Lectures: 3, Labs: 0	**Credit Hours:** 3
Prerequisites: None	
Objectives: To develop efficient literature survey, analysis, report writing and document designing skills.	
Course Outline: Overview of technical reporting, use of library and information gathering, administering questionnaires, reviewing the gathered information. Technical exposition; topical arrangement, exemplification, definition, classification and division, casual analysis, effective exposition, technical narration, description and argumentation, persuasive strategy. Organizing information and generation solution: brainstorming, organizing material, construction of the formal outline, outlining conventions, electronic communication, generation solutions. Polishing style: paragraphs, listening sentence structure, clarity, length and order, pomposity, empty words, pompous vocabulary, document design: document structure, preamble, summaries, abstracts, table of contents, footnotes, glossaries, cross-referencing, plagiarism, citation and bibliography, glossaries, index, appendices, typesetting systems, creating the professional report; elements, mechanical elements and graphical elements. Reports: Proposals, progress reports, Leaflets, brochures, handbooks, magazines articles, research papers, feasibility reports, project reports, technical research reports, manuals and documentation, thesis. Electronic documents, Linear verses hierarchical structure documents.	
Reference Material: Greenfield, T., Research Methods, Guidance for Postgraduates, Arnold, 1996, 034064629.	

Course Name: Web Fundamentals **Course Code: MSC116**	
Course Structure: Lectures: 2, Labs: 1	**Credit Hours:** 3
Prerequisites: None	

Objectives:

This course is aimed for IT professionals new to web designing. Computer science students aiming to be web developers and to target excellent web designing job market, can also benefit from this course.

Course Outline:

HTML(Hypertext Markup Language)

a) Internet, Web and HTML Fundamentals
b) What is HTML
c) The World Wide Web and Web Servers
d) Working of Web Browsers
e) HTML's Role on the Web
f) Way of launching the Web Site

Creating Static Web Pages with HTML

a) Creating a Web page and entering Text
b) Changing and Customizations
c) Display Text in List
d) Adding Graphics into Web Pages
e) Hypertext and Creating Links
f) Issuing Links with other HTML Tags

Advance HTML

a) Tables, Forms, Images
b) Frames
c) Multimedia Objects

Java Script

a) Data types
b) Control Structures
c) Object & Function
d) Event Handling

VB Script

Introduction, Data Types, Syntax, Controls, etc.

Reference Material:

1. *Thomas A. Powell, HTML The Complete Reference, 2nd Edition, McGraw Hill, 1999.*
2. *Ann Navarro, Todd Stauffer, HTML by Example, 1st Edition, Que Corp, 1999.*
3. *Andrew Wooldrige, Mike Morgan, Mona Everett, Scott J. Walter, Special Edition Using Java Script, Que Corp, 1997.*

2ND SEMESTER

Course Name: Data Structures **Course Code:MSC121**	
Course Structure: Lectures: 2, Labs: 1	**Credit Hours:** 3
Prerequisites: Object Oriented Programming	
Objectives: The course is designed to teach students structures and schemes, which allow them to write programmes to efficiently manipulate, store, and retrieve data. Students are exposed to the concepts of time and space complexity of computer programmes.	
Course Outline: Introduction to data structures; Arrays, Stacks, Queues, Priority Queues, Linked Lists, Trees, Spanning Trees, Graphs and Traversals. Recursion, sorting and searching algorithms, Shortest path algorithms, Hashing, Storage and retrieval properties and techniques for the various data structures. Algorithm Complexity, Polynomial and Intractable Algorithms, Classes of Efficient Algorithms, Divide and Conquer, Dynamic, Greedy	
Reference Material: 1. Data Structures and Algorithm Analysis in Java, 2/E **Mark Allen Weiss**, *Florida International University* ISBN-10: 0321370139 ISBN-13: 9780321370136 Publisher: Addison-Wesley 2. Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, 3/E **Robert Sedgewick**, *Princeton University* ISBN-10: 0201350882 ISBN-13: 9780201350883 Publisher: Addison-Wesley Professional	

Course Name: Computer Networks **Course Code: MSC122**	
Course Structure: Lectures: 2, Labs: 0	**Credit Hours: 3**
Course Outline: Review of basic concepts: The OSI Model, packet and circuit switching, network topology, ISDN. The TCP/IP protocol stack: IP, ARP, TCP and UDP, DNS, ICMP, Internet Addressing, Routing, IP Multicast, RSVP, Next Generation IP – Ipng, Wireless: Radio basics, Satellite Systems, WAP, current trends, Issues with wireless over TCP. Congestion Control: Control vs. Avoidance. Algorithms, Congestion in the Internet. Mobile IP, Voice over IP (VoIP), VPNs, Network Security. Management: Quality of Service (QoS), network vs. distributed systems management Protocols, web-based management	
Reference Material:	

James F. Kurose and Keith W. Ross, "Computer Networking – A Top-Down Approach Featuring the Internet", Addison Wesley.
Coulouris, Dollimore, Kindberg, "Distributed Systems – Concepts and Design", Addison Wesley.
William Stallings, "Data and Computer Communications", Prentice-Hall — Sixth Edition (for those who want to review basics of networking).

Course Name: Web Engineering	**Course Code: MSC123**
Course Structure: Lectures: 3 / Labs: 1	**Credit Hours:** 3

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

Course Outline: Introduces Web Engineering, a sub-discipline of Software Engineering whose methods and techniques are adapted to the special characteristics of Web-based applications. Also discusses the critical initial stage of the Web application development process – gathering requirements – and adapting traditional practices to the unique challenges of the Web.

Modeling Web Applications.
Introduces modeling techniques inspired by object-oriented design models found in traditional software engineering, but are adapted to the particular needs of web engineering. Topics include content, hypertext, and customization modeling; use cases, class diagrams, and sequence diagrams

Web Architectures, Application Design, and Accessibility
Reviews general architectures for Web application and technology-aware application designs. Discusses the concepts and techniques for engineering and evaluating user interfaces appropriate for a Web application's intended audience. Explores the interaction between users and the application's user interface. Special attention will be paid to Web technologies and standards available for audiences with special needs.

Client-Side Technologies
The lecture provides a brief review of CSS, cookies, and the Document Object Model. The majority of the lecture is devoted to introducing JavaScript, the language reference, and how JavaScript can be used to manipulate the DOM.

Developing Web Applications
In this lecture we discuss the web application development process on how web application projects are managed. Tasks and challenges in project management, managing teams, and risk management. Iterative development, development cycles, Rational Unified Process (RUP), Extreme Programming (XP).

Server-Side Technologies

Introduces the PHP5 scripting language. Syntax, language reference, libraries, file access, DBMS access (esp. MySQL), form processing, and other techniques will be presented/demonstrated. During the second lecture, we will walk through how to install and configure the Apache Web Server and PHP5 for a development environment, as well as discuss techniques and requirements for making the same platform operable for large-scale use.
Topics include encryption techniques (digital signatures, certificates, PKI), securing client/server interactions, vulnerabilities at the client (desktop security, phishing, etc.) and the server (cross-site scripting, SQL injections, etc.)

Testing, Operation & Maintenance
This lecture introduces and demonstrates Asynchronous Java and XML (AJAX), a technique for combining JavaScript, the HTML DOM, and server-side scripting for building highly interactive web applications.
Introduction to AJAX
Introduces and demonstrates Asynchronous Java and XML (AJAX), a technique for combining JavaScript, the HTML DOM, and server-side scripting for building highly interactive web applications.

Reference Material:
Kappel, G., Proll, B. Reich, S. & Retschitzegger, W. (2006). **Web Engineering, 1st ed.** Hoboken, NJ: Wiley & Sons. ISBN: 04700-1554-3.

Course Name: Software Engineering-I **Course Code: MSC124**	
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Object Oriented Paradigm/Programming	
Objectives: To study various software development models and phases of software development life cycle. The concepts of project management, change control, process management, software development and testing are introduced through hands-on Team Projects.	
Course Outline: Introduction to Computer-based System Engineering; Project Management; Software Specification; Requirements Engineering, System Modelling; Requirements Specifications; Software Prototyping; Software Design: Architectural Design, Object-Oriented Design, UML modelling, Function-Oriented Design, User Interface Design; Quality Assurance; Processes & Configuration Management; Introduction to advanced issues: Reusability, Patterns; Assignments and projects on various stages and deliverables of SDLC.	
Reference Material: *Software Engineering 8E by Sommerville Addison Wesley, 2006* *Software Engineering: A Practitioner's Approach /7E,* Roger Pressman, McGraw-Hill, 2009	
Course Name: Database-I Course Code: MSC125	
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Data Structures and Algorithms
Objectives: The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques. The course primarily focuses on relational data model and DBMS concepts.
Course Outline: Basic database concepts; Entity Relationship modelling, Relational data model and algebra, Structured Query language; RDBMS; Database design, functional dependencies and normal forms; Transaction processing and optimization concepts; concurrency control and recovery techniques; Database security and authorization. Small Group Project implementing a database. Physical database design: Storage and file structure; indexed files; b-trees; files with dense index; files with variable length records; database efficiency and tuning.
Reference Material: *Database Systems 8E,* C.J.Date, Addison Wesley Pub. Co. (2004). *Database Systems: A Practical Approach to Design, Implementation and Management 5E,* R.Connolly and P.Begg, Addison-Wesley Pub. Co (2009). Fundamentals of Database Systems, 5/E, Elmasri and Navathe, Addison-Wesley, ISBN: 0-201-74153-9.

Course Name: Computer Architecture	**Course Code: MSC126**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: Digital Logic and Design	
Objectives: Get a deeper understanding of how computers work, working knowledge of various subsystems and the general principles that affect their performance, analyze the performance of systems and quantify the performance measurements, fundamentals of all technologies, and advanced architectural features that boost the performance of computers	
Course Outlines: Fundamentals of Computer Design including performance measurements & quantitative principles, principles of Instruction Set Design, Operands, addressing modes and encoding, pipelining of Processors: Issues and Hurdles, exception handling features, Instruction-Level Parallelism and Dynamic handling of Exceptions, Memory Hierarchy Design, Cache Design, Performance Issues and improvements, Main Memory Performance Issues, Storage Systems, Multiprocessors and Thread Level Parallelism. Case Studies.	
Resources: *Computer Architecture: A Quantitative Approach* by Hennessy & Patterson, Morgan & Kauffman Series (2006) Fourth Edition *Computer Organization & Design* : *The Hardware/Software Interface* By Patterson & Hennessy, Morgan & Kauffman Series (2008) Fourth Edition	

3RD SEMESTER

Course Name: Operating Systems	**Course Code: MSC231**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours:** 3
Prerequisites: None	
Objectives: To help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with the layered approach that makes design, implementation and operation of the complex OS possible.	
Course Outline: History and Goals, Evolution of multi-user systems, Process and CPU management, Multithreading, Kernel and User Modes, Protection, Problems of cooperative processes, Synchronization, Deadlocks, Memory management and virtual memory, Relocation, External Fragmentation, Paging and Demand Paging, Secondary storage, Security and Protection, File systems, I/O systems, Introduction to distributed operating systems. Scheduling and dispatch, Introduction to concurrency. Lab assignments involving different single and multithreaded OS algorithms.	
Reference Material: *Applied Operating Systems Concepts*, 7th Edition, Silberschatz A., Peterson, J.L., & Galvin P.C. 2004 *Modern Operating Systems*,3rd Edition, Tanenmaum A.S., 2008.	

Course Name: Analysis of Algorithms	**Course Code: MSC232**
Course Structure: Lectures: 3 / Labs: 0	**Credit Hours:** 3
Prerequisites: Discrete Structure, Data Structures and Algorithms	
Objectives: Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.	
Course Outline: Introduction; Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms.	
Reference Material: *Introduction to Algorithms /2E,* T. H. Cormen, C. E. Leiserson, and R. L. Rivest, MIT Press, McGraw-Hill, New York, NY, 2001. Algorithms in C++; Robert Sedgewick	

Course Name: Computer Organization and Assembly Language **Course Code: BCS233**	
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Digital Logic Design	
Objectives: The main objective of this course is to introduce the organization of computer systems and usage of assembly language for optimization and control. Emphasis should be given to expose the low-level logic employed for problem solving while using assembly language as a tool. At the end of the course the students should be capable of writing moderately complex assembly language subroutines and interfacing them to any high level language.	
Course Outline: Microprocessor Bus Structure: Addressing, Data and Control, Memory Organization and Structure (Segmented and Linear Models), Introduction to Registers and Flags, Data Movement, Arithmetic and Logic, Programme Control, Subroutines, Stack and its operation, Peripheral Control Interrupts, Interfacing with high level languages, Real-time application. Objectives and Perspectives of Assembly Language, Addressing Modes, Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code, Describe actions inside the processing chip, Discuss operations performed by an instruction set, Write a fully documented program, Using an assembler of choice.	
Reference Material: • Stallings, "Computer Organization & Architecture", 7th ed, Prentice HALL, 2006. • Irvine, Assembly Language for Intel-based Computers, 5th ed, Prentice Hall, 2007. • Computer Organization and Design, The Hardware/Software Interface, 4th ed, by David A. Patterson and John L. Hennessy, 2008. Elsevier Publishers.	

Course Name: Theory of Automata and Formal languages Course Code: MSC234	
Course Structure: Lectures: 3 Labs: 0	**Credit Hours:** 3
Prerequisites: Discrete Structures	
Objectives: The course aims to develop an appreciation of the theoretical foundations of computer science through study of mathematical & abstract models of computers and the theory of formal languages. *Theory of formal languages* and use of various abstract machines as 'recognizers' and parsing will be studied for identifying/validating the synthetic characteristics of programming languages. Some of the abstract machines shall also study as 'Transducers'.	
Course Outline: *Finite State Models:* Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene's theorem, Transducers (automata with output), Pumping lemma and non regular language *Grammars and PDA:* Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs , Normal form grammars and parsing, Decidability, Chomsky's hierarchy of grammars *Turing Machines Theory:* Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Context sensitive Grammars, Defining Computers by TMs.	
Text Books/Reference Books: • An Introduction to Formal Languages and Automata, By Peter Linz, 4th edition, Jones	

& Bartlett Publishers, 2006

- Theory of Automata, Formal Languages and Computation, By S. P. Eugene, Kavier, 2005, New Age Publishers, ISBN (10): 81-224-2334-5, ISBN (13) : 978-81-224-2334-1.
- John Hopcroft and Jeffrey Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd edition, 2001, Addison-Wesley.
- Introduction to Languages and the Theory of Computation, By John C. Martin3rd edition, 2002, McGraw-Hill Professional.

Course Name: Artificial Intelligence	**Course Code: MSC235**
Course Structure: Lectures: 2 / Labs: 1	**Credit Hours:** 3
Prerequisites: Discrete Structures	

Objectives:

This course studies four main objectives of AI. Modelling the environment by constructing computer representations of the real world. Perception and reasoning - obtaining and creating information/knowledge to populate a computational representation. Taking actions by using the knowledge of the environment and desired goals to plan and execute actions. Learning from past experience.

Course Outline: Artificial Intelligence: Introduction, Intelligent Agents. Problem-solving: Solving Problems by Searching, Informed Search and Exploration, Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation. Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning, Statistical Learning Methods, Reinforcement Learning. Communicating, perceiving, and acting: Communication, Probabilistic Language Processing, Perception and Robotics. Introduction to LISP/PROLOG and Expert Systems (ES) and Applications.

Reference Material:

- Artificial Intelligence: Structures and Strategies for Complex Problem Solving: International Edition By George F. Luger, 6th edition: Pearson Education, 2008.
- Artificial Intelligence: A Modern Approach, By Stuart Jonathan Russell, Peter Norvig, John F. Canny, 2nd Edition, Prentice Hall, 2003.

Course Name: DataBase-II Course Code: MSC236	
Course Structure: Lectures: 3 / Labs: 1	**Credit Hours:** 3

Objectives: This course provides students with the technical skills required to program a database solution by using Some tool. Like Oracle or MS SQL server.

Course Outline: Concern DBMS Overview

- What Is concern DBMS?
- Its Integration
- DBMS Databases
- DBMS Security
- Working with DBMS

Overview of Programming DBMS

- Designing Enterprise Application Architecture
- DBMS Programming Tools
- The Transact-SQL Programming Language
- Elements of Transact-SQL
- Additional Language Elements
- Ways to Execute Transact-SQL Statement

Creating and Managing Databases

- Creating Databases
- Creating Filegroups
- Managing Databases
- Introduction to Data Structures

Creating Data Types and Tables

- Creating Data Types
- Creating Tables
- Generating Column Values
- Generating Scripts

Implementing Data Integrity

- Types of Data Integrity
- Enforcing Data Integrity
- Defining Constraints
- Types of Constraints
- Disabling Constraints
- Using Defaults and Rules

- Deciding Which Enforcement Method to Use

Planning Indexes

- Introduction to Indexes
- Index Architecture
- How DBMS Retrieves Stored Data
- How DBMS Maintains Index and Heap Structures
- Deciding Which Columns to Index

Creating and Maintaining Indexes

- Creating Indexes
- Creating Index Options
- Maintaining Indexes
- Introduction to Statistics
- Querying the **sysindexes** Table
- Setting Up Indexes Using the Index Tuning Wizard
- Performance Considerations

Implementing Views

- Introduction to Views
- Advantages of Views
- Defining Views
- Modifying Data Through Views
- Optimizing Performance by Using Views
- Performance Considerations

Implementing Stored Procedures

- Introduction to Stored Procedures
- Creating, Executing, Modifying, and Dropping Stored Procedures
- Using Parameters in Stored Procedures
- Executing Extended Stored Procedures
- Handling Error Messages
- Performance Considerations

Implementing User-defined Functions

- What Is a User-defined Function?
- Defining User-defined Functions
- Examples of User-defined Functions

Implementing Triggers

- Introduction to Triggers
- Defining Triggers
- How Triggers Work
- Examples of Triggers
- Performance Considerations

Programming Across Multiple Servers

- Introduction to Distributed Queries
- Executing an Ad Hoc Query on a Remote Data Source
- Setting Up a Linked Server Environment
- Executing a Query on a Linked Server
- Managing Distributed Transactions
- Modifying Data on a Linked Server
- Using Partitioned Views

Optimizing Query Performance

- Introduction to the Query Optimizer
- Obtaining Execution Plan Information
- Using an Index to Cover a Query
- Indexing Strategies
- Overriding the Query Optimizer

Analyzing Queries

- Queries That Use the AND Operator
- Queries That Use the OR Operator
- Queries That Use Join Operations

Managing Transactions and Locks

- Introduction to Transactions and Locks
- Managing Transactions
- DBMS Locking
- Managing Locks

Reference Material:
Introduction to SQL Server by Joseph

4TH SEMESTER

Course Name: Visual Programming	Course Code: **MSC241**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours:** 3
Prerequisites: None	

Objectives: The aim of this course is to teach an object oriented programming overviews and specially getting expertise in vb.net application programming. In this course students learn basics of vb.net GUI programming, which includes several built in classes like buttons, text fields etc. After completion of this course the students becomes able to understand the object oriented aspect of this language.

Course Outline:

-Installation of Visual Studio.Net
-Introduction to Vb.Net IDE
- -Data Types
  - -Byte, Short, Integer, Long, Double
  - -Char, Date
- -Declaring Variables & Constants
- -Structures
- -Operators
  - -Arithmetic (+, -, *, /, %, ^)
  - -Concatenation (&)
- -Conditions
  - -If … then…. else statement
  - -Select… end select statement
- -Loops
  - -For …next, Exit for
  - -While … end while, Exit do
- -Working with strings
  - -Toupper, Tolower, Replace, Trim, Len, StrReverse
- -Advantages of procedures

-Types of Procedures
-Scope of procedures
    -Public, Private, Friend
-Exceptions Handling
    -Try
    -Catch
    -Finally
-Introduction to classes
    -Simple class
    -Declaring objects
        -Instance and static class
        -Adding methods to class
-Encapsulation
-Inheritance
-Common properties
    -Design Vs Run time
-Name,Size, Location, Enabled, Visible,
-Backcolor, Fore color, Font,Text
-Vb.net forms
-Working with different controls in vb.net
-Textbox Control
    -Common Properties
    -Read only, Password char, Max length, Multi line,
    -Border style, Scrollbar, Text align
-Button Control
-Common Properties
-Text, Flat style, Image, Image align, Text align
-Common Properties
-Checkbox and Radio Controls
        -Appearance, Checked, Check state, Text, flat style
    -Listbox Control
        -Scrollbars, Items, Sorted, Size
    -Combobox Control
        -Items, Max length, Selected item
-Built-in Dialog Controls
    -Open file Dialog
    -Save file Dialog
    -Font Dialog
-Rich Text Box
    -Properties
        -Autosize, Borderstyle, Lines, Selectionfont, Selectioncolor Methods
            -Clear, Load file, Save file, Undo, Redo, Copy, Paste

Reference Material:

Visual Basic.net Bible By Bill Evjen, Jason Beres
Visual Basic.net Mastering series

Course Name: Computer Graphics Course Code: MSC242	
Course Structure: Lectures: 2 / Labs: 1	**Credit Hours:** 3
Prerequisites: Object Oriented Programming , Visual Programming	
Objectives: Study of various algorithms in computer graphics and their implementation in any programming language.	
Course Outline: Graphics hardware. Fundamental algorithms. Applications of graphics. Interactive graphics programming — graph plotting, windows and clipping, and segmentation. Programming raster display systems, Differential Line Algorithm, panning and zooming. Raster algorithms and software — Scan-Converting lines, characters and circles. Scaling, Rotation, Translation, Region filling and clipping. Two and three dimensional imaging geometry(Perspective projection and Orthogonal projection) and transformations. Curve and surface design, rendering, shading, colour and animation.	
Reference Material: 1. Computer Graphics, Principles and Practice, J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, Addison-Wesley ISBN: 0-201-12110-7. *2. Computer Graphics,* F.S.Hill, Maxwell MacMillan ISBN: 0-02-354860-6. 3. Interactive Computer Graphics: Functional, Procedural and Device-level methods; Peter Burger and Duncan. F. Gillies; Addison-Wesley, (2003)	

Course Name: Software Engineering-II Course Code:MSC243	
Course Structure: Lectures: 3hrs	**Credit Hours:** 3
Course Objective: The students will study techniques for software verification, validation and testing. They would also study reliability and performance issues in software design and development.	

Course Description:

Object Oriented Analysis and Design

Introduction

a) Introduction & Definitions
b) OO Modeling Concepts
c) OO Developments

Modeling as a Design Technique

a) Object Modeling Technique

Object Modeling

a) Objects & Class
b) Links & Associations
c) Generalization & Inheritance
d) Grouping Constructs
e) Aggregation
f) Abstract Class
g) Multiple Inheritance, Meta Data, Candidate Key

Dynamic Modeling

a) Events & States.
b) Operations, Nested State Diagram
c) Concurrency, Advanced Dynamic Modeling Concepts

Functional Modeling

a) Functional Models, DFD
b) Specifying Operations, Constraints
c) Relation of Functional to Object and Dynamic Model

Design Methodology

a) Methodology review
b) OMT as Software Engineering Methodology
c) OMT Methodology, Impact of OO approach

System Design

a) Overview of System Design
b) Breaking of System into Sub Systems
c) Identifying Concurrency
d) Allocating Subsystems to Processors and Tasks
e) Management of Data Store
f) Handling Global Recurs
g) Choosing Software Control Implementation
h) Handling Boundary Conditions
i) Settling Traded-off Priorities
j) Common Architectural Framework
k) Architecture of ATM System

Implementation

a) Form Design to Implementation
b) Implementation using programming languages
c) Implementation using Database System
d) Implementation using Outside a Computer

OO Testing

a) Testing OOA and Models
b) OO Testing Strategies
c) Test Case Design for OO Software
d) Testing methods applicable at class levels
e) Inter class test case design

Object Diagram Compiler

a) Background
b) Problem Statement
c) Analysis
d) System Design
e) Object Design
f) Implementation

Text Books/Reference Books:

1. *James Rumbaugh, Object Oriented Modeling and Design, 6th Edition, Prentice Hall International, 2000.*
2. *Craig Larman, Applying UML and Patterns: An introduction to Object-Oriented analysis and Design, 2nd Edition, Prentice Hall International, 2001.*
3. *James R.Rumbaugh, Michael R.Blaha, William Premerlani, Frederick Eddy, William Lorensen,Object Oriented Modeling and Design with UML, 2nd Edition, Prentice Hall, 2004.*

Course Name: Compiler Construction Course Code: **MSC244**	
Course Structure: Lectures: 2 / Labs: 1	**Credit Hours:** 3
Prerequisites: Theory of Automata and Formal Languages	
Objectives: At the end of the course students should understand the overall structure of a compiler, and will know significant details of a number of important techniques commonly used. They will be aware of the way in which language features raise challenges for compiler builders.	
Course Outline: Compiler techniques and methodology. Organization of compilers. Lexical and syntax analysis. Parsing techniques. Object code generation and optimization, detection and recovery from errors. Contrast between compilers and interpreters.	
Reference Material:	

- Compilers: Principles, Techniques, and Tools By Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Contributor Jeffrey D. Ullman ,Addison-Wesley Pub. Co., 2nd edition,1987 Original from the University of Michigan
- Modern Compiler Design, By Dick Grune, Henri E. Bal, Ceriel J. H. Jacobs, Koen G. Langendoen, John Wiley, 2000.
- Modern Compiler Implementation in C, By Andrew W. Appel, Maia Ginsburg, Contributor Maia Ginsburg, Cambridge University Press, 2004.
- Modern Compiler Design by Dick Grune, Henri E. Bal, Ceriel J. H. Jacobs, Koen G. Langendoen, 2003, John Wiley & Sons.