پوهنتون کاردان
KARDAN UNIVERSITY

Course Catalogue

For Bachelor's in Computer Science

Contents

Program Introduction

Department of Computer Science at Kardan University was established in 2006 to cover the gap of skilled personnel in public and private organizations and produce quality graduates with command on modern technologies via advanced theoretical, experimental and applied research culture. The program is implemented through a credits-based system, requiring students to complete 138 credits in four (04) years duration. Students need to quality 36 subjects, including 20-core, 12-elective, and 4-specialization subjects and final year project.

Currently, the computer science department offers courses to overcome the technological gap in networks and telecommunication, software engineering and development, programming with object-oriented concepts, a mobile application for android and web application development. To achieve excellence in these areas, students in the computer science department enjoy a conducive environment inside classrooms and computer labs. The medium of instruction is English for all courses.

Department of computer science inducted experienced and qualified faculty members to produce research-oriented graduates to compete at the national level and internationally.

Course Courses and pre-requisites

Core Courses

No.	Code	Course Name	Pre-requisite	Credits
1	CS-101	Programming Language Concepts	Nil	4
2	CS-108	Data communication and Network	Nil	4
3	CS102	Database System Concepts	Nil	4
4	CS-107	Object-Oriented Programming	Prog Lang Concepts	4
5	CS-105	Digital Logic and Design	Nil	4
6	CS-106	Principles of Software Engineering	Nil	4
7	CS-201	Data Structure and Algorithm	Object-Oriented Prog	4
8	CS-203	Modern Programming Languages	Object-Oriented Prog	
9	CS-204	Network Strategies	Data Comm	4
10	CS-202	Artificial Intelligence	Nil	4
11	CS-205	Operating System Concepts	Data Structure and Algo	4
12	CS206	Computer Organization and Assembly	Digital Logic and design	4
13	CS-308	Web Engineering	Web Fundamentals	4
14	CS-303	Analysis of Algorithm	Data structure and algo	4
15	CS-301	Mobile application in Java	Modern Prog language	4
16	CS-306	Computer Architecture	Computer organization and Assembly	4
17	CS-208	Database administration	Database System Concepts	4
18	CS-304	Automata Theory	Data Structure & algo and Programming courses	4
19	CS-401	Compiler Construction	Automata Theory	4
20	CS-402	Computer Graphics	Object-Oriented Prog, College algebra	4
Total Courses				**20**
Total Credits				**80**

Elective Courses and its Pre-Requisites

Elective Courses				
No.	**Code**	**Course Name**	**Pre-requisite**	**Credits**
1	CS-104	Introduction to IT	Nill	03
2	CS-103	Web Fundamentals	Nill	03
3	MTH-034	College Algebra	Nill	03
4	LG-003	English	Nill	03
5	CS-302	Visual Programming-I	Object-Oriented Prog	03
6	CS-207	Web Technologies	Web Fundamentals	03
7	CS-305	Software Design and Modeling	Principles of Software Engineering	03
8	CS-307	Visual Programming-II	Visual Programming-I	03
9	HIS-007	History of Afghanistan	Nill	03
10	ISS-001	Islamic Studies	Nill	03
11	GIS-047	Geographic Information System	Nill	03
12	BUS-029	E-Commerce	Nill	03
13	BUS-028	Introduction to Business	Nill	03
14	COM-031	Business Communication	Nill	03
15	COM-032	Effective Presentation Skills	Nill	03
16	MKT-035	Digital Marketing	Nill	03
17	BUS-030	Management Information System	Nill	03
18	SS-039	Introduction to Psychology	Nill	03
19	SS-040	Introduction to Sociology	Nill	03
20	EQ-043	Soft Skills	Nill	03
Total Courses		**(Out of above mentioned course, student need to select only 12)**		**12**
Total Credits				**36**

Specialization Courses and its Pre-Requisites

Specialization Courses				
No.	**Code**	**Course Name**	**Pre-requisite**	**Credits**
		Database Development (Choose 4 Courses)		16
1	CS-413	Data Ware Housing	All Core	4
2	CS-412	ORDBMS	All Core	4
3	CS-407	Distributed Databases	All Core	4
4	CS-406	Data Mining	All Core	4
		Network Administration (Choose 4 Courses)		16
1	CS-409	Wireless Networks	All Core	4
2	CS-415	WAN Technologies (CCNA)	All Core	4
3	CS-408	Network System & Programming	All Core	4
4	CS-414	Telecommunication Systems	All Core	4
		Total Courses 4		
		Total Credits		**16**
		Software Engineering (Choose 4 Courses)		
1	CS-410	Software Project Management	All Core	4
2	CS-416	Software Quality Assurance	All Core	4
3	CS-417	Formal Methods	All Core	4
4	CS-411	Advance Topics in software Engineering	All Core	4
		Total Courses 4		
		Total Credits		**16**

Thesis				
No.	**Code**	**Course Name**	**Pre-requisite**	**Credits**
1		Undergraduate Thesis		6
Total Courses				**1**
Total Credits				**6**

Summary			
No.	**Category**	**Credits**	**Percentage**
1	Core	80	58%
2	Specialization	16	12%
3	General	36	26%
4	Thesis	6	4%
Total		**138**	**100%**

Individual Course Descriptions

Core Courses

Programming Language Concept	
Course Code	CS-101
Credits	4
Pre-requisite	None
Description	The first practical programming subject that the students study in the first semester is programming language concepts. In this module, the students know how programming is different from that of any application and packages. The students study and learn the language's grammatical rules and then utilize that knowledge and skill for learning the object-oriented concepts to develop a small scale application.
Course Objectives	– Explain how an existing C++ program works. – Discover errors in a C++ program and describe how to fix them. – Critique a C++ program and describe ways to improve it. – Analyze a problem and construct a C++ program that solves it. – Choose and apply the required commands to develop C++ programs in an Integrated Development Environment.
Learning Resources	– Object-Oriented Programming in C++ , 4th Edition by Robert Lafore (Text Book), – C++ How to Program, 3rd Edition Deitel (Ref. Book)
Course Outlines (Week-wise)	
1	**Chapter 1: Introduction** 1.1 Introduction to Programming language 1.2 Why we study this subject? 1.3 History of C++ 1.4 Compiler 1.5 Interpreter 1.6 What is a program? 1.7 Structure of C++ programs 1.8 C++ Phases 1.9 Preprocessor Directives 1.10 Header file
2	1.11 The main() function 1.12 C++ statements 1.13 Keywords 1.14 Tokens 1.15 Variables 1.16 Rules for writing variable names 1.17 Declaration of variables 1.18 Initialization of variables 1.20 Constants

3	**Chapter 2: Data Types** 1.1 Data Types 1.2 Why we use data types 1.3 Basic data types 1.4 Derived data types
4	1.5 Integer constant 1.6 Floating-point constant 1.7 Character constant 1.8 String constant
5	**Chapter 3: Operators and Decision Control Structure** 3.1 Operators 3.2 Arithmetic operators 3.3 Relational Operators 3.4 Logical Operators
6	3.4 If-Statement 3.5 Single if statements 3.6 Compound if statements
7	**Chapter 4:** 4.1 if-else statement 4.2 Single if-else statement 4.3 Compound if-else statement
8	4.4 Nested if-else statement 4.5 Compound nested if-else statements
9	**Chapter 5:** 5.1 The switch statement 5.2 Break statement 5.3 Continue statement
10	5.4 The go-to statement 5.5 Drawbacks of go-to statement 5.6 Advantages of using switch statement
	Mid Term
11	**Chapter 6: Loop Control Structure** 6.1 loop 6.2 For-loop 6.3 Single statements inside body of loop 6.4 Compound statements inside body of loop 6.5 Advantages of for-loop
12	6.6 The while-loop 6.7 For-loop vs. while-loop 6.8 Which to use when? 6.9 do-while-loop
13	**Chapter 7: Arrays** 7.1 What is an Array? 7.2 Why we use it? 7.3 How can we use it? 7.4 Subscripts in array

14	7.5 Memory showing an array. 7.6 Accessing Array Elements 7.7 Examples
15	**Chapter 8: Multidimensional Array** 8.1 What is a multidimensional array 8.2 Rows and columns in a multidimensional array 8.3 The memory map of 2D array
16	8.4 Accessing 2D array elements 8.5 Mentioning the number of rows is optional 8.6 Practical Examples
17	**Chapter 9: Functions** 9.1 Function definition 9.2 Function prototype 9.3 Function call
18	9.4 Built-in and user define functions 9.5 Arguments or parameters 9.6 Pass by value 9.7 Pass by address

Database System Concepts

Course Code	CS-102
Credits	4
Pre-requisite	None
Description	This course is designed for an introduction to database management. It provides students with the essential concepts, principles, and techniques of modern database systems from a user perspective. This means that the lecture focuses on the functionalities offered by database systems and not on the methods to implement them. Specifically, the course teaches students the ability to develop a solution to a real-world data management problem that requires applying the theories and practices developed in class. From a theoretical point of view, this course covers the essential principles for designing, analyzing, and using computerized database systems.
Course Objectives	– Knowledge of DBMS, both in terms of use and implementation – Knowledge of DBMS design – To Learn the Methodologies of Database Analysis – Experience with SQL – Experience working as part of a team – Experience with analysis and design of (DB) software
Learning Resources	– Modern Database Management Jeffery A. Hoffer, Mary B. Prescott, Prentice Hall – Database Systems Principles, Design and Implementation Catherine Ricardo, Maxwell Macmillan

Course Outlines (Week-wise)	
1	– Learning Goals – Introduction

	– Basic Concepts and Definitions – Data – Information – Data Versus Information – Metadata
2	– Database Management Systems – Data Models – Entities – Relationships – Relational Databases
3	– Traditional File Processing Systems – Disadvantages of File Processing Systems – The Database Approach – Advantages of The Database Approach – Components of the Database Environment – The Range of Database Applications
4	– Sample E-R Diagram – E-R Model Notation – Modeling Entities and Attributes – Entities – Entity Type Versus Entity Instance – Entity Type Versus System Input, Output, or User – Strong Versus Weak Entity Types – Naming and Defining Entity Types
5	– Attributes – Required Versus Optional Attributes – Simple Versus Composite Attributes – Single-Valued Versus Multivalued Attributes – Stored Versus Derived Attributes – Identifier Attribute – Naming and Defining Attributes – Modeling Relationships – Basic Concepts and Definitions in Relationships – Attributes on Relationships – Associative Entities
6	– Degree of a Relationship – Unary Relationship – Binary Relationship – Ternary Relationship – Cardinality Constraints – Minimum Cardinality – Maximum Cardinality – A Ternary Relationship – Modeling Time Dependent Data – Multiple Relationships
7	– Introduction – The Relational Data Model – Basic Definitions

	– Relational Data Structure – Relational Keys – Properties of Relations – Removing Multivalued Attributes from Tables
8	– Steps in Normalization – Functional Dependencies and Keys – Determinants – Candidate Keys – Normalization Example – Step 0: Represent the View in Tabular Form – Step 1: Convert to First Normal Form – Remove Repeating Groups – Select the Primary Key – Anomalies in 1NF – Step 2: Convert to Second Normal Form
	Mid Term
9	– Removing Transitive Dependencies – Determinants and Normalization – Step 4: Further Normalization
10	– Introduction of Oracle 11g – Installation of Oracle 11g – The working environment in Oracle 11g – Introduction to SQL *PLUS – Introduction to SQL Developer – SQL Statements
11	– Data Retrieval Language – Restricting and Sorting Data – List the capabilities of SQL SELECT statements – Generate a report of data from the output of a basic SELECT statement – Select All Columns – Select Specific Columns – Use Column Heading Defaults – Use Arithmetic Operators
12	– Write queries that contain a WHERE clause to limit the output retrieved – List the comparison operators and logical operators that are used in a WHERE clause – Describe the rules of precedence for comparison and logical operators – Use character string literals in the WHERE clause – ORDER BY Clause
13	– What is DDL? – Creating a Simple Table – Managing Tables Data Types
14	– Integrity Constraints – Domain Constraints – Types of Constraints – NOT NULL

	– UNIQUE – FOREIGN KEY – CHECK Constraints – Entity Integrity – Referential Integrity – Creating Relational Table – Well-Structured Relations
15	– Step I: Map Regular Entities – Composite Attributes – Multivalued Attributes – Step 2: Map Weak Entities – Step 3: Map Binary Relationships – Map Binary One-to-Many Relationships – Map Binary Many-to-Many Relationships – Map Binary One-lo-One Relationships
16	– Step 4: Map Associative Entities – Identifier Not Assigned – Identifier Assigned – Step 5: Map Unary Relationships – Unary One-ta-Many Relationships – Unary Many-ta-Many Relationships – Step 6: Map Ternary (and n-ary) Relationships – Step 7: Map Super-type/Sub-type Relationships

Data Communication and Networks	
Course Code	CS-108
Credits	4
Pre-requisite	None
Description	This subject introduces the students to the basic concepts of networking and data communication. It also teaches the know-how required for advanced subjects like Network Strategies and specialization subjects.

Object-Oriented Programming	
Course Code	CS-107
Credits	3
Pre-requisite	Programming Language Concepts
Description	Object-oriented programming is a core subject where the students learn Object-Oriented techniques to remove structured programming limitations and provide more security to programming code. This subject will enhance students' programming skills and allow them to learn and understand Object-Oriented software design. This subject covers all characteristics of the Object-oriented paradigm.
Course Objectives	This course will cover object-oriented concepts and provide a strong base for designing. It also provides a strong base for understanding the Modern Programming languages and .Net techniques.
Learning Resources	- Object-Oriented Programming 5th Edition Robert Lafore (textbook) - Object-Oriented Programming 3rd Edition Dietle and Deitel (reference book)
Course Outlines (Week-wise)	
1	- What is OOP? - Characteristics of OOP. - The Inline function - #define Macro - The inline function Vs Macro - Drawbacks associated with Macro - Comparison of speed and Memory using inline function
2	- Storage class of variables - Automatic Storage class - The initial default value, storage, scope and lifetime of automatic SC. - Register SC - The initial default value, storage, scope and lifetime of Register SC. - Static SC - The initial default value, storage, scope and lifetime of static SC. - SSC variable value persists between different function calls
3	- The External SC - The initial default value, storage, scope and lifetime of External SC. - ESS global nature - Recursion - Steps in Recursive procedure - Expensive nature of Recursion in term of time and memory - Function overloading

	- Number, Type and order of arguments
4	- Structure - Structure data members and memory map - Accessing data members using the dot operator - Array of structure - Pointer to structure - Nested structure - Structure members by default are public
5	- Classes and Objects - Private and public member access specifiers - Data members and member functions - Object and memory map - Constructor and Destructor - Constructor Overloading - The Default Constructor - The default copy Constructor - Two styles of Constructor
6	- Operators - Operator Overloading - Unary Operator Overloading - Binary Operator Overloading - The new Operator - The delete Operator - The this Pointer - The void pointer
7	- Inheritance - Protected Member access Specifier - The base class - Derived classes - Public Inheritance - Private Inheritance - Protected Inheritance - Multiple Inheritance - Multiple Level of Inheritance - Default Inheritance
8	- What is Polymorphism - The Virtual function - The Pure Virtual function - Upcasting - Dynamic Binding - Static Binding - The abstract class - The concrete derived classes - Function overriding

9	- The V-Pointer - The V-Table - The Scope Resolution Operator - The setw() Manipulator - The typecasting - The enumerators - Types of errors and its causes

Database Administration	
Course Code	CS-208
Credits	4
Pre-requisite	Database System Concepts
Description	The purpose of this course is to introduce you to the concepts and procedures of Database Administration. It is assumed that you are studying for, or are in, a Database Administrator position. Therefore the emphasis will be on how to use the information for database creation and administration. You will likely find that we cover material that you have been exposed to in other program courses, such as Database, programming, and administration.
Course Objectives	After successfully completing this course, students will be able to: - Understand the role of a database management system in an organization. - Understand basic database concepts, including the structure and operation of the relational data model. - This course is related to Database Administration, such as the physical realization of the Database, including physical database design and implementation, security and integrity control, maintenance of the operational system, and ensuring satisfactory performance of users' applications. - Construct simple and moderately advanced database queries using Structured Query Language (SQL) in ORACLE. - Design and implement a database project using Oracle - Understand the concept of a database transaction and related database facilities. - Understand the role of the database administrator.
Learning Resources	- OCA Oracle Database 11g: SQL Fundamentals I - Lecture slides and Handout
Course Outlines (Week-wise)	
1	**Introduction to Database Administration** Database DBMS and RDBMS What is Database Administration? Duties of DBA Installing DBMS software Tools to access Database

	Introduction to Oracle Discuss the basic design, theoretical and physical aspects of a relational database Categorize the different types of SQL statements Describe the data set used by the course Log onto the Database using the SQL Developer environment Log onto the Database using the SQL Plus environment Save queries to files and use script files in SQL Developer
	SQL Capability Data Retrieval Language Restricting and Sorting Data List the capabilities of SQL SELECT statements Generate a report of data from the output of a basic SELECT statement Select All Columns Select Specific Columns Use Column Heading Defaults Use Arithmetic Operators
	SQL Projection SQL Projection Comparison Operators Logical Operators Understand Operator Precedence Table Structure Meta Data DESCRIBE command to display the table structure
	Restricting and Sorting Data Write queries that contain a WHERE clause to limit the output retrieved List the comparison operators and logical operators that are used in a WHERE clause Describe the rules of precedence for comparison and logical operators Use character string literals in the WHERE clause ORDER BY Clause Sort output in descending and ascending order
	Single-Row Functions and Group Function Single Row Function Describe the differences between single row and multiple row functions Manipulate strings with character function in the SELECT and WHERE clauses Manipulate numbers with the ROUND, TRUNC and MOD functions Describe implicit and explicit data type conversion Date Data Type Function Apply the NVL
	Group Function Aggregate of Data Different GROUP BY clause Function Using the DISTINCT Keyword Group Functions and Null Values HAVING Clause Restricting Group Results using HAVING Clause Nesting Group Functions

Displaying Data from Multiple Tables Obtaining Data from Multiple Tables Qualifying Ambiguous Column Names Types of Joins Natural joins: NATURAL JOIN clause USING Clause Using Table Aliases with the USING Clause ON Clause
Displaying Data from Multiple Tables Using Table Aliases with the USING Clause ON Clause for Joining of Data Applying Additional Conditions to Join
Manipulating Data Data manipulation language SELECT Statement INSERT Statement UPDATE Statement DELETE Statement MERGE Statement
Transaction control Transaction control Language COMMIT ROLLBACK SAVEPOINT TRUNCATE Statement Advantages of COMMIT and ROLLBACK statements Explicit Transaction Control Statements Implicit Transaction Processing
SubQuery Single-row and multiple-row sub-queries SET operators Use a SET operator to combine multiple queries into a single query Control the order of rows returned when using the SET operators
Database Integrity and DDL What is DDL? Creating a Simple Table Managing Tables What is Database Integrity? Types of Constraints NOT NULL UNIQUE PRIMARY KEY FOREIGN KEY CHECK Constraints
Alter Table and Flashback Creating a Table Using a Subquery ALTER TABLE Statement Read-Only Tables Dropping a Table DROP TABLE … PURGE

FLASHBACK TABLE Statement
Controlling User Access System Privileges Creating new users Removing users Removing tables User System Privileges Granting System Privileges What Is a Role? Creating and Granting Privileges to a Role Changing Your Password Object Privileges Passing On Your Privileges Revoking Object Privileges
Oracle Logical and Physical Storage Structure What is Physical Storage Structure? Data Files and Redo Log Files Control File and Archive Log Files Password File Parameter File (PFile and SPFile) What is Logical Storage Structure? Table-spaces and Segments Extents and Oracle Blocks Creating Table-spaces
Oracle Instance Components of Oracle Server What is Oracle Instance? Memory Structure (SGA and PGA) Parts of SGA User and Server Process Background Processes Starting up and Shutdown Database Terminating the User Sessions

Principles of Software Engineering	
Course Code	CS-106
Credits	4
Pre-requisite	None
Description	This course mostly covers the technical aspects of software development. Its main focus is to educate the student s with software designing, coding, and testing phases. It will cover software data design, software architecture, software component-level design and software user interface design along with software testing strategies. Students will not only study but will also apply these practices for the project. Class activities will help to adopt these skills for real-world projects.
Course Objectives	The students will - Understand latest up to date methodologies in system/software engineering - Understand the application of different fact-finding techniques - Understand how to develop a formal and best fit solution to the organization systems - Be able to successfully complete the system/software projects using the practitioner's approach for system/software development
Learning Resources	Textbook: Software Engineering by Roger S. Pressman, 5th Edition, McGraw-Hill Reference book Software Engineering by Ian Sommerville, 8th Edition
Course Outlines (Week-wise)	
	1. Overview and introduction of the course 2. Software Engineering: The process 3. Process models I 4. Process models II 5. Fact-finding techniques I 6. Fact-finding techniques II 7. Fact-finding techniques III 8. Requirement analysis 9. Developing a Software Requirement Specification (SRS) document 10. Analysis modeling 11. Data modeling 12. Function modeling 13. Behavior modeling 14. User interface modeling 15. The software design I 16. The software design II

Digital Logic & Design	
Course Code	CS-105
Credits	3
Pre-requisite	Introduction to Computer
Description	The course has been designed to meet the following objectives. Ability to perform conversions among decimal, binary, octal and hexadecimal number systems. Ability to analyze and design gate-level combinational logic circuits using Boolean algebra. Ability to analyze, design, and utilize combinational components such as adders, multiplexers, decoders and encoders. Ability to analyze and design simple synchronous sequential circuits. Ability to design registers and counters.
Course Objectives	Will enable the students to: Understand digital machines/computers Understand the procedure on how to implement a specified objective into digital machines/computers Understand the logic required to implement a specified circuit. Understand different combinational circuit and their design Understand different sequential circuits and their design.
Learning Resources	Text Book: Digital Logic and design by Moris Mano Reference book: Digital Logic and design by Malvino
Course Outlines (Week-wise)	
1	Number system I
2	Number system II
3	Number system III
4	Boolean algebra I
5	Boolean algebra II
6	Boolean algebra III
7	Logic gates I
8	Logic gates II
9	Circuit design
10	Circuit simplification techniques
11	Combinational circuits I
12	Combinational circuits II
13	Combinational circuits III
14	Sequential circuits I

15	Sequential circuits II
16	Sequential circuits III

Web Engineering	
Course Code	CS-308
Credits	3
Pre-requisite	Web Fundamentals
Description	This course will teach PHP, providing both a solid understanding of the fundamentals and a sense of where to look for more advanced information. Through demonstrations and real-world examples, this course provides the knowledge you need to begin building dynamic Web sites and Web applications using PHP.

Data Structure & Algorithms	
Course Code	ET-2505
Credits	3
Pre-requisite	Object Oriented Programming
Description	This subject talks about how to organize the data at the time of development of software so that all operations on data become easy, efficient and effective. The operations includes, insertion of new records, deletion of existing data, sorting, searching, merging and traversing of data that already exist. In this subject the students will study the solution for creating different structures of data in the computer memory.
Course Objectives	
Learning Resources	Data Structure (Text Book) Schaum's Outline Series, 4th Edition Data Structure theoretical and Practical approach (Reference Book) By Nazar Muhammad, 2nd Edition
Course Outlines	
1	• What is Data Structure? • Linear and Non Linear Data Structures • Different Operations on DS • What is Algorithm? • How to write algorithm • The Pseudo Codes • Characteristics of an algorithm • Execution flow • Sequential, Selection and Iterative

	• Algorithm for Multiple alternative and loops • Implementation of the proposed algorithms
2	• Starting Linear Data Structure • Array and its types • Accessing Array elements using Dope Vector method for one D-array • Accessing Array elements using Dope Vector Method using 2-D array • The LIFFE Access method for 2-D array • Advantages and drawbacks of both Methods
3	• What is stack? • Stack applications • Push Algorithm and its implementation • Pop Algorithm and its implementation • What is Queue? • Queue applications • Push algorithm and its implementation • Pop algorithm in Queue and its implementation
4	• What is DEQUEUE? • Push algorithm and its implementation • Pop algorithm and its implementation • Different applications of DEQUEUE • Comparison of QUEUE and DEQUEUE • Sorting Procedures • Bubble Sort technique • Bubble Sort algorithm and implementation
5	• Selection Sort • Selection Sort Mechanism • Algorithm and implementation • Insertion Sort • Insertion Sort technique • Algorithm and implementation • Comparison of Selection and Insertion sort techniques in term of speed and Memory
6	• What is Quick Sort • Solution of different examples using Quicksort technique • Algorithm and Implementation • Radix Sort • The solution of different examples using Radix Sort technique • Algorithm and implementation • Comparison of Quick Sort and Radix Sort in term of Speed and Memory

	• Comparison of all Sorting techniques • Applications in existing software
7	• What is Searching? • What is Linear Search technique? • Linear Search Algorithm and Implementation • What is Binary Search? • Binary Search algorithm and implementation • Comparison of both Searching mechanisms in term of speed
8	• What is tree? • Tree Terminologies. • Building tree using Offline and Online methods. • What is a binary tree? • Strictly and Complete binary trees. • Building binary tree from an expression • Traversing binary tree. • In order, pre-order and post order • Infix, prefix and postfix notations • Conversion from infix to prefix and postfix using tree method. • Conversion from infix to prefix and postfix using direct method.

Automata Theory	
Course Code	CS-304
Credits	4
Pre-requisite	Programming Language Concepts
Description	The mathematical description of computational networks, the mechanical computation limitations, and the formal specification of languages are highly interrelated disciplines, and all require a great deal of mathematical maturity to appreciate. A computer science undergraduate is often expected to deal with all these concepts, and so, this course attempts to make it possible for average students by developing the standard mathematical models of computation devices, as well as investigating the cognitive and generative capabilities of such machines
Course Objectives	
Learning Resources	1. Introduction to Computer Theory, Daniel I.A. Cohen (Text Book) 2. Introduction to Automata Theory, Languages and Computation, John E. Hopcroft, Rajeev Motwani (Ref Book)
Course Outlines (Week-wise)	
1	Introduction to Course Title and Objectives • Why study Automata Theory? • Alphabets • Strings

	• Null string, words, valid and in-valid alphabets • Length of a string • Reverse of a string Defining Languages • descriptive definition of languages and its examples
2	Recursive Definitions • Defining languages through recursive definitions, Kleen star closure, plus operation
3	Regular Expressions • Defining languages through regular expressions
4	Applications of Regular Expressions • Practical examples Introduction to Finite Automata • definition of finite automata • Transition table • Transition diagram
5	Finite Automata • Practical examples • Vending machine • C/C++ comments recognizer • String matching machine Transition Graph • Transition Graph vs Finite Automata
6	Kleen's Theorem • TG to Regular Expression • Regular Expression to FA • Nondeterministic FA • NFA and Kleene's Theorem
7	Finite Automata with Output • Moore Machine • Mealy Machine
8	Pushdown Automata • Pushdown Automata • Applications of PDA
9	Turing Machine • Insertion machine • Deletion machine

Computer Organization & Assembly Language	
Course Code	CS-206
Credits	3
Pre-requisite	Digital Logic & Design
Description	This course introduces to the students the basic architecture of the Intel Processors and the computer's working. Hardware programming can be done in Assembly language, so the second part is to cover the basics of assembly language programming and viewing the results of assembly programming on the hardware. The aim is not only to focus on the programming but also to introduce to students the processor's internal operations along with the memory and other related hardware.
Course Objectives	- Learn about assembly language, its uses and its advantages. - Know about the role of assembly language in hardware. - Learn about Basic microcomputer architecture. - Learn about the internal structure of the processor. - Know about the working of a computer and the Load decode and execute processes. - Learn to develop programs using assembly language, - Understand the interface between running programs and the microprocessor - Discovering the mysteries of how the CPU chip in your computer works. Learn to develop small applications for IA-x86 architecture.
Learning Resources	- Assembly Language for Intel-Based Computers By Kip R. Irvine - Google docs Class Lectures, Lab work and the internet.
Course Outlines (Week-wise)	
1	Introduction to the Course Assembly Language Applications The History of PC Assemblers Goals and required background Assembly, machine and high level languages
2	Why learn assembly language Assembly language programming tools Data representation
3	Processor Architecture Basic Microcomputer Design

	Instruction Execution Cycle Reading from Memory
4	How Programs Run Load and Execute Process Multitasking
5	IA-32 Processor Architecture 41 Modes of Operation Basic Execution Environment Address Space Basic Program Execution Registers Floating-Point Unit Other Registers 6Intel Microprocessor History
6	I7A-32 Memory Management R8eal-address Mode 20-bit Linear Address Calculation Protected Mode
7	Basic Elements of Assembly Language Integer Constants Integer Expressions Real Number Constants Character Constants String Constants Reserved Words Identifiers
8	Directives Instructions Assembling, Linking, and Running Programs The Assemble-Link-Execute Cycle
Mid Term	
9	Chapter 6 :Defining Data Data Definition Statement Defining BYTE and SBYTE Data Multiple Initializers
10	Defining DWORD and SDWORD Data Defining QWORD Data
11	Data Transfer Instructions Operand Types Direct Memory Operands Copying Smaller Values to Larger Ones
12	Chapter 8 :Addition and Subtraction ADD Instruction SUB Instruction
13	JMP and LOOP Instructions JMP Instruction LOOP Instruction Copying a String
14	Multiplication and Division Instructions MUL Instruction IMUL Instruction

	DIY Instruction Signed Integer Division
15	Defining and Using Procedures PROC Directive Defining a Procedure Example: Sum of Three Integers

Operating System Concepts	
Course Code	CS-205
Credits	4
Pre-requisite	Data Structure and Algorithm, Object-Oriented programming
Description	This course teaches the basic Operating System abstractions, mechanisms, and their implementations. The course's core focuses on OS support for resource management (CPU, memory, I/O), processes, scheduling, deadlocks, memory management, and file systems. UNIX and Windows NT are general-purpose operating systems used as examples when studying these concepts. Laboratory assignments of process, process communication, and file systems are given.
Course Objectives	This course will help the students to - Understand the basic concepts of operating system - Understand the advanced concepts of operating system - Understand a different kind of memories and their management - Understand virtual memory and their management - Understand other courses i.e. computer architecture, programming languages etc.
Learning Resources	Text Book: Operating Systems Design and Implementation by Andrew S. Tanenbaum and Albert S. Woodhull, Third Edition, Prentice Hall Reference book: Operating System Concepts by Abraham Silberschartz, Peter Baer Galvin and Greg Gagne, 7th Edition by John Wiley & Sons Inc.
Course Outlines (Week-wise)	
1	Overview and introduction of the course
2	Background
3	Operating system structures
4	Process
5	Process II
6	Process scheduling
7	Process scheduling II
8	Process scheduling III

9	Memory management
10	Memory management
11	Virtual memory I
12	Virtual memory II
13	Linux I
14	Linux II
15	Linux III
16	Linux IV

Analysis of Algorithms	
Course Code	CS-303
Credits	4
Pre-requisite	Data Structure and Algorithm
Description	This course provides a rigorous introduction to worst-case asymptotic algorithm analysis. It develops a classical graph and combinatorial algorithms for such problems as sorting, shortest paths and minimum spanning trees. It also Introduces the concept of computational intractability and NP-completeness.
Course Objectives	- Analyze the worst-case running time of an algorithm as a function of input size - Solve Recurrence relations - Understand and implement Divide and Conquer strategy - Understand the concepts of Dynamic programming - Understand the concepts of Greedy Algorithm - Understand the concepts of Graph traversing - Understand and explain the basics of Complexity theory.
Learning Resources	- Introduction to Algorithms, by T. Cormen, C. Leiserson, and R. Rivest. - Lecture slides and Handouts
Course Outlines (Week-wise)	
1	Introduction to Algorithm 1.1 Introduction 1.2 Origin of word Algorithm and Definition 1.3 Analyzing Algorithms 1.4 Model of Computation 1.5 Brute-Force Algorithm 1.6 Running Time Analysis
2	Mathematical Algorithms and Growth Function 2.1 Mathematical Algorithms 2.2 Asymptotic Notation 2.2 Asymptotic Upper Bound 2.4 Asymptotic Lower Bound 2.5 Asymptotic Tight Bound 2.6 Asymptotic Behavior
3	Analysis of Iterative & Recursive Algorithms

	3.1 Iterative Algorithms 3.2 Analysis of Iterative Algorithms 3.3 Examples of Iterative Algorithms
4	Analysis of Iterative & Recursive Algorithms 3.1 Recursive Algorithms 3.2 Analysis of Recursive Algorithms 3.3 Examples of Recursive Algorithms
5	Divide and Conquer Strategy 5.1 Introduction to Divide and Conquer Strategy 5.2 Merge Sort 5.3 Introduction of Merge Sort 5.4 Algorithm of Merge Sort 5.5 Analysis of Algorithm 5.6 Run Time Calculation
6	Hash Table 6.1 Direct Addressing 6.2 Hash Table 6.3 Examples of Hashing 6.4 Hash Functions 6.5 Collision
7	Hashing II 7.1 Linear Probing 7.2 Quadratic Probing 7.3 Linked List chaining 7.4 Open Addressing 7.5 Hashing Animation 7.6 Applications of Hashing 7.7 When Hashing is Suitable
8	Graph Searching 8.1 Introduction to Graph Searching 8.2 Techniques of Graph Searching 8.3 Breadth-First Search 8.4 Example of Breadth-First Search
9	Depth First Search 9.1 Introduction to Depth First Search 9.2 Example of Depth Search 9.3 Analysis of Depth First Search
Mid Term	
10	Minimum Spanning Tree 10.1 Introduction to MST 10.2 Growing a Spanning Tree 10.3 Examples of Spanning Tree 10.4 Applications of MST 10.5 Prim's Algorithm
11	Greedy Algorithms 11.1 Local Optimal Solution 11.2 Global Optimal Solution 11.3 Greedy Algorithms 11.4 Examples of Greedy Algorithms 11.5 Examples: Counting Money

12	Greedy Algorithms (Kruskal's Algorithm) 12.1 Introduction to Kruskal's Algorithm 12.2 Examples of Kruskal's Algorithm 12.3 Analysis of Kruskal's Algorithm
13	Greedy Algorithms(Huffman Encoding Algorithm) 13.1 Introduction to Huffman Encoding Algorithm 13.2 Encoding Correctness, 13.2 Encoding Activity Selection
14	Dynamic Programming 14.1 Dynamic Programming 14.2 Edit Distance Algorithm 14.3 Edit Distance Applications
15	Dynamic Programming II 15.1 Different Dynamic Programming Algorithms and their Analysis
16	Dynamic Programming III 16.1 Chain Matrix Multiply 16.2 Chain Matrix Multiply Examples Analysis and Calculation of Running Time
17	Dijkstra's Algorithm 17.1 Dijkstra's Algorithm 17.2 Correctness of Dijkstra's Algorithm 17.3 Analysis of Dijkstra's Algorithm

Compiler Construction	
Course Code	CS-401
Credits	4
Pre-requisite	Automata Theory
Description	This course will discuss the major ideas used today to implement programming language compilers, including lexical analysis, parsing, syntax-directed translation, abstract syntax trees, types, and type checking, intermediate languages, and dataflow analysis program optimization, code generation, and runtime systems. As a result, you will learn how a program written in a high-level language designed for humans is systematically translated into a program written in low-level assembly more suited to machines. We will also touch on how programming languages are designed, programming language semantics, and why there are so many different kinds of programming languages.
Course Objectives	Students should be able to: - Know about the internal working of a compiler. - Analyze the internal structure of the compiler and also discuss the different parts. - Practice and enhance their critical thinking. - Practice reflective thinking by participating in class discussions, answering questions, class activities, and practicing development. Students will also

	display reflective thinking by practicing problems, homework and practical assignments.
Learning Resources	- Compilers, Principles, Techniques & Tools Alfred V. Aho, Ravi Sethi (Text Book) - Modern Compiler Design, D. Grune, eta (Reference Book)
Course Outlines (Week-wise)	
1	Introduction Language Processors The Structure of a Compiler The Evolution of Programming Languages
2	The Science of Building a Compiler Applications of Compiler Technology Programming Language Basics
3	A Simple Syntax directed Translator Introduction Syntax definition Parsing
4	Lexical analysis Symbol table Intermediate Code Generation
5	Chapter 3: Lexical Analysis The role of lexical analyzer Specification of tokens
6	Finite automata Writing RE for tokens Design of a Lexical-Analyzer Generator
7	**Chapter 4 : Issues in Compiler Construction** **1.1** Issues in compiler construction **1.2** The Role of the Parser **1.3** Representative Grammars **1.4** Syntax Error Handling **1.5** Error-Recovery Strategies File Handling **1.6** File handling to implement a component of a compiler
8	**Chapter 5: Use of Regular Expressions in the Development of Compiler.**
Mid Term	
9	**Chapter 6 : Parsing** **2.1** Top-down parsing **2.2** How top-down parsing works
10	**2.3** Left recursion & its elimination **2.4** Derivation
11	**2.5** Bottom-up Parsing **2.6** LL(k) parsing **2.7** Shift reduce parsers
12	**Chapter 7 : Syntax-Directed Translation** **4.1** Syntax-Directed Definitions **4.2** Evaluation Orders for SDD's
13	**4.3** Applications of Syntax-Directed Translation **4.4** Syntax-Directed Translation Schemes
14	**Chapter 8 : Intermediate-Code Generation** **5.1** Variants of Syntax Trees

	5.2 Three-Address Code **5.3** Types and Declarations **5.4** Translation of Expressions **5.5** Type Checking
15	**Chapter 9 : Runtime Environments** **6.1** Storage Organization **6.2** Access to Nonlocal Data on the Stack **6.3** Heap Management
16	**6.4** Introduction to Garbage Collection **6.5** Introduction to Trace-Based Collection **6.6** Short-Pause Garbage Collection

Computer Graphics	
Course Code	CS-402
Credits	4
Pre-requisite	Object Oriented Programming
Description	Computer Graphics is the illustration field of Computer Science. Today, its use spans all scientific fields virtually and is utilized for design, presentation, education, and training. Computer Graphics and its derivative, visualization, have become the primary tools by which the flood of information from Computational Science is analyzed. This course is not a discussion of standard graphics systems, nor experimentation with canned packages. We will focus on the fundamental algorithms of computer graphics from which all these other packages are.
Course Objectives	This course will cover the concepts, techniques and algorithms in the development and design of graphical software. In this course, we emphasized the resolution of the display units and discussed the control mechanisms through recent and modified algorithms. The course covers the motion of images depends upon solid mathematical models.
Learning Resources	Computer graphics 2nd Edition By Hern and Packer
Course Outlines (Week-wise)	

Week 1, 2	• What is Computer graphics • History and background • Different applications • Basic primitives • Picture elements • Resolution • High definition System • Aspect Ratio • Bit map System and pix Map System • Basic Colors • Colors Combination • Dot pitch • Built in graphical libraries and I/O hardware
Week 3, 4	• Line and Line Segments • The slope-intercept algorithm • Alternate approach of slope-intercept algorithm • Comparison of two algorithms • Drawbacks of slope-intercept algorithm • Digital differential analyzer algorithm • Case-I • Case-II
Week 5, 6	• Drawbacks of DDA • The Bresenham's Mid-point line algorithm • Case-I • Case-II • Case-III • Circle generating procedures • Simple Circle algorithm • Drawbacks of Simple Circle algorithm • Bresenham's Midpoint Circle algorithm • Case-I with examples • Case-II with examples • Case-III with examples
Week 7, 8	• Scan Converting Ellipse Procedure • Region-I • Case-I with examples • Case-II with examples • Case-III with examples • Region-II • Case-I with examples • Case-II with examples • Case-III with examples
Week 9, 10	• Transformation • Coordinate Transformation • Geometric Transformation • Types of Geometric Transformation • Translation • Matrix Representation with examples • Rotation

	• Matrix Representation with examples • Scaling • Matrix Representation with examples
Week 11, 12 Quiz No. 2	• Homogenous coordinates and 3D representation of 3D transformation • Compound Translation • Compound Rotation • Compound Scaling • With examples
Week 13, 14	• What is window and world window • Screen and viewports • Window to Viewport Mapping • Calculating the X-Coordinate • Calculating the Y-Coordinate • Examples
Week 15, 16	• What is Clipping • Point Clipping • Line Clipping • Cohen Sutherland Clipping algorithm • Trivially accepted lines • Trivially Rejected lines • Partial acceptance and rejection • Point of intersection on Horizontal boundary • Point of intersection on Vertical boundary • Different examples
Week 17, 18	• What is Projection? • Parallel projection • Orthographic PP projection • Axonometric OPP projection • Isometric, diametric and trimetric OPP projection • Oblique PP projection • Cavalier and Cabinet OPP Projection • Perspective Projection • One point, two Point and Three-Point PP

Computer Architecture	
Course Code	CS-306
Credits	4
Pre-requisite	Computer organization and Assembly, Digital logic and Design
Description	The main focus will be to educate the students about the computer system's multiple functional units and its working. The student will learn its design, implementation and working. Multiple computer architectures will be studied analytically. Parallel processing will educate the students with the current

	computer architecture styles in the market and enable them to design parallel programs for multiprocessing environments.
Course Objectives	- The student will learn different functional units of the computer - The student will understand different architectural style for the computer system - The student will understand the art of how they can design their own functional units for the computer system - The student will understand advanced approaches in computer architecture - The student will be able to work in different manufacturing units of the electronic machine's production.
Learning Resources	Text Book: - Computer Organization and Architecture by William Stalling Reference books - Digital logic and design by Malvino - Digital logic and design by Moris Mano
Course Outlines (Week-wise)	
1	Overview and introduction of the course
2	Computer architecture: The IAS computers: A case study
3	Computer architecture: The IAS computers: A case study
4	Decoding and encoding
5	The computer memory I
6	The computer memory II
7	Computer memory design
8	Multiplexing and DE multiplexing
9	The processor I
10	The processor II
11	Reduced Instruction Set Computer RISC
12	Complex Instruction Set Computer CISC
13	Parallel processing I
14	Parallel processing II
15	Parallel processing III
16	Parallel processing IV

Artificial Intelligence

Course Code	CS-202
Credits	4
Pre-requisite	Nill
Description	The purpose of this course is to introduce you to the concepts and procedures of Artificial Intelligence. It will cover simple representation schemes, problem-solving paradigms, constraint propagation, and search strategies. Areas of

	application such as knowledge representation, natural language processing, expert systems and computer vision will be explored. Another part of the course is to introduce ProLog programming and its environment and its application in the knowledge representation scheme of Artificial Intelligence. The use of logic for problem-solving will also be practiced upon.

BCS Specialization Courses – Object Oriented Relational Database Management System

ORDBMS	
Course Code	CS-412
Credits	4
Pre-requisite	Database System Concepts & Database Administration
Description	This course is designed for practicing Oracle professionals who have basic experience with SQL and the use of a relational database. Experience using SQL with a relational database is highly desirable. By the end of this course, the student will be able to formulate advanced SQL queries, including correlated sub-queries and outer joins. The student will also become familiar with the internals of Oracle11g/12c SQL and will be able to use the EXPLAIN PLAN utility to tune SQL statements.

Course Objectives	After successfully completing this course, students will be able to: - Understand the role of a database management system in an organization. - Understand basic Object-Oriented Relational Database. - Features of Object-Orientation, Object-Oriented data models, - Object-Relational DBMSs, Object-Oriented Database Management Systems (OODBMSs), Features of OODBMSs, - Different OODBMSs, Research issues in OODBs. - Understand the concept of a database transaction and related database facilities. - Know about some leading object-oriented database management systems. - Know about research issues in object-oriented databases.
Learning Resources	- Object-Oriented Database Systems: Approaches and Architectures - Oracle Database 11g PL/SQL Programming - OCA Oracle Database 12C: SQL Fundamentals I - Lecture slides and Handout
Course Outlines (Week-wise)	
1	**Introduction to SQL Query and ORDBMS** Retrieving Data using SELECT Restricting Data Sorting Data
2	**Introduction to Sub Query** Writing Sub-queries Guidelines for using Sub-queries Types of Sub-queries
3	**Creating Schema Objects-1** Using DDL to Create and manage Tables Implementing Constraints
4	**Creating Schema Objects-2** Creating Views Types of Views Sequences Indexes
5	**PL/SQL and Interacting with Oracle Server** What is Procedural Language (PL)? Types of PL/SQL Blocks Features of PL/SQL Basic Block Structure of PL Execution of PL/SQL
6	**DML Statements in PL/SQL** Writing SELECT statement in PL/SQL Declaring Variables Writing DML statements in PL/SQL Control Transactions in PL/SQL
7	**PL/SQL Control Structure** What are Control Structures? Conditional Structures Looping Structures
8	**PL/SQL Cursors**

	What are Cursors? Types of Cursor Cursor Attributes Working with Record Cursor with Parameter
9	**Exception Handling Part-1** What are Exceptions? Types of Exceptions Writing PL/SQL Block to handle Exceptions Trap Predefined and User-defined Exceptions

Distributed Databases	
Course Code	CS-407
Credits	4
Pre-requisite	Database System Concepts & Database Administration
Description	The purpose of this course is to introduce you to the concepts and procedures of the Distributed Database. The use of distributed systems has become a common practice in today's computing environment, especially with the internet's easy access. However, distributed Database Systems (DDBSs) are generally implemented in relatively large organizations and need a better understanding of the database and networking concepts. The same two concepts provide the foundation for this course. The emphasis is on the design and management issues of DDBS and the implementation issues.

BCS Specialization Courses – Computer Networks

Wireless Networks	
Course Code	CS-409
Credits	4
Pre-requisite	Data Communication and Network, Network Strategies
Description	The course of Network Strategies contains all basic and medium levels of concepts and practical illustrations required to establish an organizational network. The contents have been carefully selected and standardized with world class syllabi, which will equip the students with the latest conceptual and practical learning related to network design deployment, troubleshooting, and future enhancements.

WAN Technologies	
Course Code	CS-415
Credits	4
Pre-requisite	Network Strategies
Description	This course will cover the key concepts and techniques in designing and constructing Local Area Network, Virtual LAN, and Wide Area Network in detail. In this course, the students will learn IPv4 and IPv6 Network designing and deployment.
Course Objectives	- To implement small to medium size networks - To implement, configure and troubleshoot routed networks - To identify a security threat to a network

Learning Resources	- CCNA Routing and Switching by Todd Lammle - Cisco ICND Cisco Press - www.cisco.com - Slides Provided By Instructor - Internet
Course Outlines (Week-wise)	
1	- Internetworking Basics - Internetworking Models - OSI Reference Model
2	- Ethernet Networks - Ethernet Cabling - Data Encapsulation - Cisco Three-Layer Hierarchical Model
3	- Introducing TCP/IP - TCP/IP and the DoD Model - IP Addressing - IPV4 Address Types
4	- Subnetting Class A - Subnetting Class B - Subnetting Class C
5	- VLSM Design for class A - VLSM Design for class B - VLSM Design for class C
6	- IOS User Interface - Switch/Router Components - Command Line Interface - Administrative Configurations - Router and Switch Interfaces - Viewing saving erasing configurations
7	- Backup and Restoring IOS Configurations - Configuring DHCP - Cisco Discovery Protocol - Telnet - SSH - Checking network connectivity and troubleshooting
8	- Routing Basics - IP Routing Process - Configuring IP Routing - Types of IP Routing - Configuring static routes
9	- Dynamic Routing

	- Routing Information Protocol RIP V1 - Routing Information Protocol RIP V2
10	- OSPF Basics - Configuring OSPF - OSPF and Loopback Interfaces - Verifying OSPF Configuration
11	- EIGRP Basics - Configuring EIGRP - Verifying EIGRP Configuration
12	- Switching Basics - Configuring Catalyst Switches - MAC-Address Tables
13	- VLAN Basics - Identifying VLANs - Configuring VLANs - Inter VLANs Routing
14	- Access Control List - Standard ACL - Extended ACL - Configuring ACL on Routers
15	- Network Address Translation - How NAT Works - Types of NAT - Configuration of NAT on Cisco Routers
16	- PPP - HDLC - Practical configuration

Network & System Programming

Course Code	CS-408
Credits	4
Pre-requisite	Object-Oriented Programming, Data Communication and Network
Description	The course of Network and System Administration contains all basic and medium levels of concepts and practical illustrations required to establish an organizational network. The contents have been carefully selected and standardized with world class syllabi, which will equip the students with the latest conceptual and practical learning related to network design deployment, troubleshooting, and future enhancements.

Telecommunications	
Course Code	CS-414
Credits	4
Pre-requisite	Data Communication and Network, Network Strategies
Description	This course provides the student with an understanding of the evolution of telecommunication networks from traditional Public Switched Telephone Network (PSTN), through the emergence of data networks, local area networks, Integrated Services Digital Network (ISDN), broadband ISDN, Frame Relay, ATM, Cellular Networks, Routing Protocols, and Signaling System 7
Course Objectives	- To introduce the students to the broad area of telecommunication - To know the working mechanism of the telecommunication system - To understand the field of mobile communication
Learning Resources	- Introduction to Telecommunication Network Engineering, 2nd Edition, by T. Aattalainen - Fundamentals of Telecommunication Networks, T. Saadawi, Wiley USwww.cisco.com - Slides provided by Instructor
Course Outlines (Week-wise)	
1	- What is Telecommunication - Significance of Telecommunication - History of Telecommunication
2	- Standardization - Standards Organization - National Standardization Authorities - European Organizations

	- American Organizations - Global Organizations
3	- Basic Telecommunication Network - Operation of Conventional Telephone - Signaling to the Exchange from the Telephone - Telephone Numbering
4	- Switching and Signaling - Telecommunication Network - Virtual Private Networks - INs - PSTN - DCN - TMN
5	- Types of Information and Their Requirements - Simplex, Half Duplex and Full Duplex - Frequency and Bandwidth - Analog and Digital Signals - Advantages of Digital Technology - Analog signals over Digital Networks
6	- PCM - Sampling - Quantizing - Quantizing noise - Binary coding - PCM encoder and decoder
7	- Adaptive PCM - Differential PCM - DM - Adaptive DPCM - Speech coding of GSM - Power level of signals and Decibels
8	- Transmission - Basic Elements of Transmission - Signal and Spectra - Radio Transmission - AM - FM - PM - Antennas
10	- Maximum data rate of a transmission channel - Multiplexing - FDM - TDM

	- PCM Frame Structure - SDH and SONET
11	- Transmission Media - Copper Cables - Optical Fiber Cables - Radio Transmission - Satellite Transmission
12	- Transmission Equipment - Modems - Terminal Multiplexers - Add/Drop Multiplexers - WDM - Optical Amplifiers - Microwave Relay Systems
13	- Mobile Communication - Cellular Radio Principles - Structure of Cellular Network - HLR and VLR - Radio Channels - MS in Idle mode - Outgoing call - Incoming call - Handoff - MS Transmission Power
14	- GSM - Structure of GSM Network - Physical channel - Logical channel - Operations of GSM Networks - GSM Enhanced Data Services
15	- Data communication principles - Computer communications - Serial and Parallel data communication - Circuit and Packet Switching
16	- ISDN - DSL - Cable TV Networks - Wireless Access - Fiber Cable Access - Leased lines and WANs
17	- Frame Relay - ATM - Protocol Layers of ATM

- Cell structure of ATM
- Physical Layer of ATM
- Switching of ATM Cells
- Applications and Future of ATM