III/IV B.Tech IT Syllabus of Admitted Batch 2015-2016

UG PROGRAM – B.TECH (IT)

W.E.F. Admitted Batch 2015-2016

DEPARTMENT OF INFORMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES
(UGC AUTONOMOUS)

(Affiliated to Andhra University, Approved by AICTE & Accredited by NBA)
Sangivalasa, Bheemunipatnam Mandal, Visakhapatnam Dt. 531162.
Phone: 08933-225083/84/87 Fax:226395
Website: www.anits.edu.in email:principal@anits.edu.in

SEMESTER-1

INTELLECTUAL PROPERTY RIGHT AND PATENTS

IT311(A) **Credits: 3**
Instruction: 3 Periods & 1 Tut /week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite: Not required.

Course Objectives:

- Understand intellectual property rights and law of copy rights.
- Procedure to apply various patents for innovative ideas and products.
- Aware various trade laws in the field of business.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Understand the importance of Intellectual property rights and its usage.
2.	Know various International laws in trade policies.
3.	Identify the international trade secrets trade secretes litigation
4.	Analyze international trade and copy right laws

SYLLABUS

UNIT I: (Text Book 1) **7 Periods**
Introduction to Intellectual Property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT II: (Text Book 1) **10 Periods**
Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting and evaluating trade mark, trade mark registration processes.

UNIT III: (Text Book 1 & 2) **15 Periods**
Law of copy rights: Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copy right registration, notice of copy right, international copy right law.
Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT IV: (Text Book 1) **10 Periods**
Trade Secrets: Trade secretes law, determination of trade secretes status, liability for misappropriations of trade secrets, protection for submission, trade secretes litigation.
Unfair Competition: Misappropriation right of publicity, False advertising

UNIT V: (Text Book 2) **18 Periods**
New development of intellectual property: New developments in trade mark law; copy right law, patent law, intellectual property audits. International overview on intellectual property, international-trade mark law, copy right law, international patent law and international development in trade secrets law.

Text Books:

1. Deborah, E. Bouchoux, Cengage learning, Intellectual property right,
2. Prabuddha Ganguli, Intellectual property right – Unleashing the knowledge economy, Tata Mc Graw Hill Publishing Company Ltd.

DATA STRUCTURES

IT311 (B) **Credits: 3**
Instruction: 3 Periods & 1 Tut /week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:
C Programming.

Course Objective:

- Assess how the choice of data structures impacts the performance of programs.
- Choose the appropriate data structure and algorithm design method for a specified application.
- Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, binary search trees, and graphs and writing programs for these solutions.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Compare and contrast the benefits of dynamic and static data structures: linked lists and arrays
2.	Evaluation of infix, prefix and postfix expressions and conversion between infix, prefix and postfix using stacks. Implement linear data structure Queue using arrays and linked lists.
3.	Implement sorting and searching techniques and analyze their computational complexity worst, average and best in terms of the size of the list(n)
4.	Solve and analyze time complexities of shortest path problem using non linear data structures trees and graphs with Prims and Krushkals algorithms and Dijkstra's & Warshall's algorithms.

Course matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	3					1	1	1		1	3	2
CO2	1	1	2					1	1	1		1	3	2
CO3	2	2	3					1	1	1		1	3	2
CO4	2	3	2					1	1	1		1	3	2

SYLLABUS

UNIT-I: Introduction **12 Periods**

Introduction to data structures, arrays and structures. Dynamic Memory Management, Abstract Data Type (ADT).

List: Definition and examples- Primitive Operations- Representation using array and Linked List. Types of Linked Lists and implementation: single, double and circular. The array and linked list advantages, disadvantages and applications.

UNIT-II: Stacks and Queues **12 Periods**

The Stack ADT: Definition, Primitive Operations and representation. Stack ADT implementation using array. Applications of Stacks: Prefix, infix and Postfix notations, conversion between infix, prefix and postfix, postfix evaluation using stacks. Recursion: definition and examples (ex: Towers of Hanoi Problem, other examples).

Queue ADT: Definition, Primitive operations and Representation. Queue ADT implementation using array. Circular Queue Operations and implementation using array. The queues applications.

UNIT-III: Sorting and Searching **12 Periods**

Sorting: General background, selection sort, bubble sort, insertion sort, merge sort.

Searching: General background, linear search, binary search .

Introduction to Hashing, Hash Function, Hashing techniques, Collision Resolution Methods: Open Addressing, Chaining.

UNIT-IV: Trees **12 Periods**

Trees: Introduction, Terminology, Binary trees: Terminology, Representation of binary tree using Arrays and Linked list. Tree Traversal Techniques, applications of binary tree-binary search tree, Expression tree. Node deletion from a binary search tree. conversion of a general tree to binary tree.

UNIT-V: Graphs **16 periods**

Graphs: Introduction- terminology, Representation of graphs-linked list and adjacency matrix, Warshall’s Algorithm for path matrix,Applications of graphs: single source shortest path-Dijkstra’s algorithm, Minimum Spanning Tree algorithms: Prims and Krushkals.

Graph traversals-Breadth-First Search, Depth-First Search.

Text Book:

1. Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structure, computer science Press.

Reference Books:

1. Y. Langsam, M. Augenstin and A. Tannenbaum, “Data Structures using C” Pearson Education, 2nd Edition, 1995.
2. Richard F, Gilberg, Forouzan, Cengage,”Data Structures”, 2/e, 2005.

SOFTWARE ENGINEERING CONCEPTS

IT311 (C)	**Credits: 3**
Instruction: 3 Periods & 1 Tut /week	Sessional Marks: 40
End Exam: 3	End Exam Marks: 60

Prerequisite:

Computer fundamentals, any programming language.

Course Objectives:

- Understand software process models such as waterfall and evolutionary models etc.
- Understand software requirements and prepare SRS document.
- Understand software testing approaches such as unit testing and integration testing etc.
- Understand quality control and ensure good quality software.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	To understand basics of software engineering
2.	Apply suitable process model for a software project
3.	Gather requirements to Analyze and design a software project
4.	Understand testing concepts

SYLLABUS

UNIT - I: 10 Periods

Introduction: Evolution and impact of Software Engineering, Software Development Projects, Exploratory style of software, Emergence of Software Engineering, Changes in software development, computer Systems Engineering

UNIT - II: 10 Periods

Software life cycle models: Use of Life cycle model, classical waterfall model, iterative waterfall model, prototyping model, evolutionary model, spiral model, comparison of different life cycle models

UNIT – III: 10 Periods

Requirements analysis and specifications: Requirement gathering and analysis, Software requirements Specification, Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4 GL

UNIT - IV: 10 Periods

Software Design: Outcome of design process, Characteristics of a good software design, Cohesion and coupling, Layered arrangement of module, Approaches to software design, Object oriented versus function oriented design approaches

UNIT - V: **8 Periods**

Coding and testing: Coding, Code review, Software documentation, Testing, Testing in the large versus testing in the small Unit testing, Black box testing, White box testing, Debugging, Program analysis tools, Integration Testing Object oriented Programming, System testing, General issues associate with testing

Text Books:

1. Rajib Mall, Fundamentals of Software Engineering, PHI, third edition

Reference Books:

1. Waman S Jawadekar, Software Engineering: A Primer, Tata McGraw-Hill, 2008
2. R. J. Leach, Introduction to Software Engineering, CRC Press.

DATABASE MANAGEMENT SYSTEMS

IT312 **Credits: 3**

Instruction: 3 Periods & 1 Tut/Week Sessional Marks: 40

End Exam: 3 Hours End Exam Marks: 60

Prerequisite:

Fundamentals of computers, knowledge in any program language

Course Objectives:

- Understand basic database concepts, including the structure and operation of the relational data model.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).
- Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.
- Understand the concept of a database transaction and related database facilities, including concurrency control, backup and recovery, locking and protocols.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Design the database system for the database application using the ER model.
2.	Construct Relational Model from ER Model and formulate queries using SQL for extracting information from database.
3.	Optimize the database design by applying normalization principles on databases
4.	Examine the serilizability of non serial schedules, and database recovery using ARIES and compare the different concurrency control protocols.

Mapping of course outcomes with program outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	3					1	1	1		1	3	2
CO2	1	2	3	1				1	1	1		1	3	2
CO3	2	2	3	2				1	1	1		1	3	2
CO4	2	3	1					1	1	1		1	3	2

SYLLABUS

UNIT-I: **10 Periods**

Introduction to DBMS: Overview, File system vs. DBMS, Advantages of DBMS, Structure of DBMS, Levels of Data Abstraction, Database Users and Administrators,

E-R model: Entities, Attributes and Entity sets, Relationship and Relationship sets, Features of ER model, Conceptual database design with ER model.

UNIT-II: **8 Periods**

Relational model: Integrity constraints over relations and enforcement, Querying relation data, Logical database design, views, destroying/altering tables and views. Relational Languages: algebra and calculus

UNIT-III: **10 Periods**

SQL: Basic SQL, Query, union, interest, except, Nested Queries, Aggregated Operation, Null values, Embedded SQL, cursors, Database connectivity(ODBC and JDBC), Triggers and Active database, designing active databases

UNIT-IV: **10 Periods**

Normalization: Introduction To Schema Refinement - Problems Caused By Redundancy, Decomposition, Functional Dependency, Closure of a Set of FDs, Normal forms(First, Second, Third normal forms, BCNF, Fourth & Fifth normal forms)

UNIT-V: **10 Periods**

Transaction management: Transaction concept, transactions and schedules, concurrent execution of transactions Concurrency control: Lock management, specialized locking techniques, concurrency control without locking Crash Recovery: Aries, recovering from a system crash

Text Books:

1. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, 3rd Edition, McGraw-Hill, 2003.

Reference Books:

1. Silberschatz, Korth and Sudharshan, “Data Base System Concepts”, 5th Edition, McGrawHill, 2006.
2. Elmasri, Navathe, “Fundamentals of Database Systems”, 5th Edition, Pearson Education, 2007.

UNIX NETWORK PROGRAMMING

IT313 **Credits: 3**
Instruction: 3 Periods & 1Tut/Week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:
Fundamentals of Computer Networks, C Programming language

Course Objectives:

- Introduce Network Programming covering TCP, and UDP connections
- Explain Socket programming to design client- server environment
- Explain inter process communication consisting of pipes, FIFOs, Semaphores and message Queues

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Get familiar with the variety of interfaces and frameworks for network applications
2.	Get the knowledge of Interfaces, STREAMS, sockets, RPC libraries.
3.	Know the underlying mechanisms to program client-server model.
4.	Using UNIX socket system calls to manage multiple I/O streams

Mapping of course outcomes with program outcomes:

		PO												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	2			3							3	2	2	3
	2	2			3	1						3	2	3	2
	3	2			3							3	2	2	3
	4	2			3							3	2	2	3

SYLLABUS

UNIT-I: 10 Periods

Introduction to Network Programming: OSI model, Unix standards, TCP and UDP, TCP connection establishment and termination, Buffer sizes and limitations, Standard Internet services, Protocol usage by common internet applications.

UNIT-II: 10 Periods

Sockets: Address structures, Value – result arguments, Byte ordering and manipulation functions and related functions. Elementary TCP sockets – socket, connect, bind, listen, accept, fork and exec functions, concurrent servers, close function and related functions.

UNIT-III: 10 Periods

TCP Client Server example: Introduction, TCP Echo server and client functions, Normal startup and Termination, Signal handling, Server process termination, Crashing and Rebooting of server host, Shutdown of server host. I/O MULTIPLEXING: I/O Models, select function, Batch input, shutdown function, poll function, TCP Echo server.

UNIT-IV: 10 Periods

Elementary UDP sockets: Introduction, recvfrom and sendto functions, UDP Echo server and client functions, Lost datagrams, Lack of flow control with UDP, determining outgoing interface with UDP, TCP and UDP echo server using select.

Elementary name and address conversions: DNS, gethostbyname function, Resolver option, gethostbyname2 function and IPV6 support, uname function, getserverbyname and getservbyport functions, other networking information.

UNIT-V: 8 Periods

IPC: Introduction, File and record locking, Pipes, FIFOs, streams and messages, Message queues, Semaphores, Shared memory.

Remote login: Terminal line disciplines, Pseudo-Terminals, Terminal modes, Control Terminals, RPC Transparency Issues.

Text Books:

1. W.Richard Stevens, UNIX Network Programming Sockets API, Volume I, 3rd Edition, PHI, 2010.

Reference Books:

1. T Chan, UNIX Systems Programming using C++, 1st Edition, PHI, 2010.
2. Graham Glass, King abls, UNIX for Programmers and Users, 3rd Edition, Pearson Education, 2010.
3. M.J. Rochkind, Advanced UNIX Programming, 2nd Edition, Pearson Education, 2008

FORMAL LANGUAGES AUTOMATA THEORY

IT314 **Credits: 4**

Instruction: 4 Periods & 1 Tut /Week Sessional Marks: 40

End Exam: 3 Hours End Exam Marks: 60

Prerequisites: The students are expected to have a strong background in the fundamentals of discrete mathematics (symbolic logic, set, induction, number theory, summation, series, combinatorics, graph, recursion, basic proof techniques, etc.), algorithms and data structures

Course Objectives:

- Understand basic properties of formal languages and formal grammars.
- Understand basic properties of deterministic and nondeterministic finite automata
- Understand the relation between types of languages and types of finite automata
- Understanding the Context free languages and grammars, and also Normalizing CFG.
- Understanding the minimization of deterministic and nondeterministic finite automata.
- Understand basic properties of Turing machines and computing with Turing machines.
- Understand the concept of Pushdown automata and its application.
- Know the concepts of tractability and decidability, the concepts of NP-completeness and NP-hard

Course outcomes:

After completion of this course, a student will be able to:	
1.	Analyze Regular, context free, context sensitive and recursively enumerable languages using Chomsky hierarchy grammars (Type-0, 1, 2, 3) based on production parameters.
2.	Discriminate the Regular grammar and context free grammar using G=(N,∑,P,S) and reconstruction of grammars based on production parameters.
3.	Design Finite automata, Pushdown automata and Turing machine for a given regular, context free and unrestricted grammar.
4.	For a given grammar, predict whether the grammar is regular or context free using pumping lemma and check whether they are closed under union, concatenation and closure. Analyze whether the given recursively enumerable language has a solution or not.

Mapping of course outcomes with program outcomes:

	O1 pp	O2 nal	O3 des	O4 inv	PO5 tools	PO6 engg	PO7 prof	PO8 ethics	O9 eam	PO10 comm	O11 econ	O12 LL	SO1	SO2
)-1	2	3	2		2		1	1				1	2	2
)-2	1	2	3	2	2		1	1				1	2	3
)-3	2	2	3	3	2	2	1	1			2	1	2	3
)-4	1	2	3	2	3	2	1	1			2	1	3	3

SYLLABUS

UNIT I: **14 Periods**

Fundamentals: Set, Representation of set, operations on set, Principle of mathematical induction, Strings**,** Alphabets, languages, operations on strings and languages**.**

Finite Automata and Regular Expressions: Basic Concepts of Finite State Systems, Deterministic and Non-Deterministic Finite Automata, Finite Automata with e-moves, Regular Expressions, Minimization of Finite Automata, Mealy and Moore Machines, Two-Way Finite Automate.

UNIT II: **14 Periods**

Regular Sets & Regular Grammars: Basic Definitions of Formal Languages and Grammars, Regular Sets and Regular Grammars, Closure Properties of Regular Sets, Pumping Lemma for Regular Sets, Decision Algorithm for Regular Sets, Myhill-Nerode Theorem, Minimization of Finite Automata.

UNIT III: **12 Periods**

Context Free Grammars and Languages: Context Free Grammars and Languages, Derivation Trees, Simplification of Context Free Grammars, Normal Forms, Pumping Lemma for CFL, closure properties of CFL"s, Decision Algorithm for CFL.

UNIT IV: **12 Periods**

Push Down Automata and Deterministic CFL: Informal Description, Definitions, Push-Down Automata and Context free Languages, Parsing and Push-Down Automata.

UNIT V: **12 Periods**

Turing Machines and Undecidability: Design and Techniques for Construction of Turing Machines, Undecidability of PCP. Chomsky Hierarchy, Regular Grammars, Unrestricted Grammars, Context Sensitive languages, Relationship between classes of languages.

Text Books:

1. E.Hopcraft & Jeffery D.Ulman, Introduction to Automata Theory, Languages & Computation, J– Narosa Publishing Company.

Reference Books:

1. Mishra & Chandra Sekharan, Theory of Computer Science, PHI.
2. Peter Linz, An Introduction To Formal Languages and Automata, 3e, Narosa Publishing House.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

IT315 **Credits: 3**

Instruction: 3 Periods & 1 Tut /week Sessional Marks: 40

End Exam: 3 Hours End Exam Marks: 60

Prerequisite:

Object oriented concepts, C++ programming

Course Objectives:

- To understand object oriented programming concepts, and apply them in problem solving.
- To learn the basics of java Console and GUI based programming.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyzeprogramming techniques to implement OOP concepts
2.	Design programs for multithreading and exception handing to create new applications
3.	design programs to implement the concepts of Java Files, collections and database in real time problem solving
4.	Develop GUI Applications

Mapping of course outcomes with program outcomes:

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	PSO1	PSO2
CO1	2	2	3	3	3				3		3	2	2	2
CO2	2	3	3	3	3				3		3	2	2	3
CO3	2	3	3	3	3				3		3	3	2	3
CO4	2	3	3	3	3				3		3	3	2	3

SYLLABUS

UNIT-I: **10 Periods**

Introduction: Introduction to Java, JVM, java garbage collector, Command line arguments, classes and objects.

Inheritance: Inheritance hierarchies, super and sub classes, Member access rules, super keyword, preventing inheritance: final classes and methods, the Object class and its methods.

Polymorphism: dynamic binding, method overriding, abstract classes and methods.

Interfaces: Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interfaces.

Inner classes: uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples.

UNIT-II: **10 Periods**

Packages: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

Exception handling: Dealing with errors, benefits of execution handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, re throwing exceptions, exception specification, built in exceptions, creating own exception sub classes.

Multithreading: Difference between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication, procedure consumer pattern.

UNIT-III: **12 Periods**

Collection Framework in Java: Introduction to Java Collections, Overview of Java Collection frame work, Generics, Commonly used Collection classes Array List, Vector, Hash table, Stack, Enumeration, Iterator, String Tokenizer, Random, Scanner, calender and Properties

Files: streams - byte streams, character streams, text input/output, binary input/output, random access file operations, File management using File class.

Connecting to Database: JDBC Type 1 to 4 drives, connecting to a database, querying a database and processing the results, updating data with JDBC.

UNIT-IV: **8 Periods**

GUI Programming with Java: The AWT class hierarchy, Introduction to Swing, Swing vs, AWT, Hierarchy for Swing components, Containers - JFrame, JApplet, JDialog, JPanel, Overview of some swing components Jbutton, JLabel, JTextField, JTextArea, simple swing applications, Layout management - Layout manager types - border, grid and flow

UNIT –V: **8 Periods**

Event handling: Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Examples: handling a button click, handling mouse events, Adapter classes.

Applets: Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet, passing parameters to applets, applet security issues.

Text Books:

1. Herbet Schidt and Dale Srien, Java Fundamentals - A comprehensive Introduction, TMH.

Reference Books:

1. P.J. Deitel and H.M. Deitel, Java for Programmers, Pearson education
2. P.J. Deitel and H.M. Deitel, Java: How to Program, PHI.

DATABASE MANAGEMENT SYSTEMS LAB

IT316 **Credits: 2**

Practical: 3 Periods/week Sessional Marks: 50

End Exam: 3 Hours End Exam Marks: 50

Prerequisite:

Fundamentals of computers, knowledge in any program language

Course Objectives:

- Understand basic database concepts, including the table structure and operations on tables.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Create and manipulate Relational Database using SQL
2.	Write PL/SQL programs, Triggers and Cursors

Mapping of course outcomes with program outcomes:

		PO												PSO	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	3	2	3		1	2	2			1		2	3	2
	2	3	2	3					1					3	2

Recommended Systems/Software Requirements:

- Intel based desktop PC
- Mysql /Oracle latest version Recommended

LIST OF EXPERIMENTS

1. DDL,DML,TCL, DCL commands
2. Creating users - roles and Granting privileges.
3. Built in functions in oracle (String - Date - Aggregate functions etc.)
4. Implement Integrity Constraints. (Key constraints - Domain constraints)
5. Implementing joins - sub queries - nested and correlated nested queries
6. Design ER Model for a given application & Convert ER model to Relational Model
7. Working with set comparison operators & views
8. Working with aggregate functions, GROUP BY, HAVING clauses
9. Implementing operations using PL/SQL blocks
10. Exception handling

11. Implementing cursors
12. Implementing triggers
13. Implementing functions and stored procedures & functions
14. Implementing packages
15. Implementing the concepts of Rollback - commit and checkpoints
16. Database connectivity

Note: Students must submit database design of a case study.

Reference Books:

1. Raghurama Krishnan, Johannes Gehrke, "Data base Management Systems", 3rd Edition, TATA McGrawHill, 2008.
2. Silberschatz, Korth, "Data base System Concepts", 6th Edition, McGraw Hill, 2010.
3. C.J.Date, "Introduction to Database Systems", 7th Edition, Pearson Education, 2002.

UNIX NETWORK PROGRAMMING LAB

IT317 **Credits: 2**
Practical: 3 Periods/week Sessional Marks: 50
End Exam: 3 Hours End Exam Marks: 50

Prerequisite:

Fundamentals of Computer Networks, C Programming language

Course Objectives:

- Introduce Network Programming covering TCP, and UDP connections
- Explain Socket programming to design client- server environment
- Explain inter process communication consisting of pipes, FIFOs, Semaphores and message Queues

Course Outcomes:

After completion of this course, a student will be able to:	
1.	write, execute and debug c programs which use Socket API
2.	understand the use of client/server architecture in application development
3.	Design reliable servers using both TCP and UDP sockets

Mapping of course outcomes with program outcomes:

		PO												PSO	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	2	3	3	3	3	3	3	3	3		3		1	2
	2	3	3		2	3	3	3	3	3		3		3	1
	3	3	3	3	1	3	3	3	3	3		3		3	2

LIST OF EXPERIMENTS

1 Implement the following forms of IPC. a) Pipes b) FIFO

2 Implement file transfer using Message Queue form of IPC.

3 Write a Program to create an integer variable using Shared Memory concept and increment the variable simultaneously by two processes. Use Semaphores to avoid Race conditions.

4 Design TCP iterative Client and Server application to reverse the given input sentence.

5 Design TCP concurrent Client and Server application to reverse the given input sentence.

6 Design TCP Client and Server application to transfer file.

7 Design a TCP concurrent Server to convert a given text into upper case using multiplexing system call “select”.

8 Design a TCP concurrent Server to echo given set of sentences using Poll functions.

9 Design UDP Client and Server application to reverse the given input sentence.

10 Design UDP Client Server to transfer a file.

11 Design using Poll Client Server application to multiplex TCP and UDP requests for converting a given text into upper case.

12 Design a RPC application to add and subtract a given pair of integers.

Reference Books:

1. Advance Unix Programming Richard Stevens, Second Edition Pearson Education
2. Advance Unix Programming, N.B. Venkateswarlu, BS Publication

JAVA PROGRAMMING LAB

IT318 **Credits: 2**
Practical: 3 Periods/week Sessional Marks: 50
End Exam: 3 Hours End Exam Marks: 50

Prerequisite:
Object oriented concepts, C++ programming language

Course Objectives:

- To design and apply object oriented programming concepts in problem solving.
- To learn the basics of java Console and GUI based programming.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Demonstrate OOP and Java programming in problem solving
2.	Design programs forreal world problems using Java Files and collections.
3.	Apply multithreading and evaluate exception handing to create new applications
4.	Develop simple and complex UI applications using GUI components and databases.

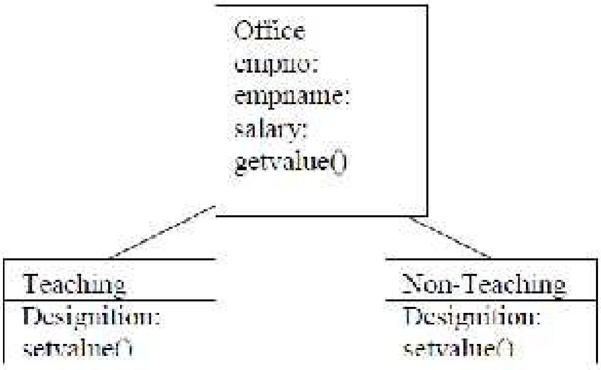
Mapping of course outcomes with program outcomes:

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	PSO1	PSO2
CO1	2	2	3	3	3				3	2	3	2	2	2
CO2	2	3	3	3	3				3		3	2	2	3
CO3	3	3	3	3	3				3		3	3	2	3
CO4	2	3	3	3	3				3		3	3	2	3

LIST OF THE EXPERIMENTS

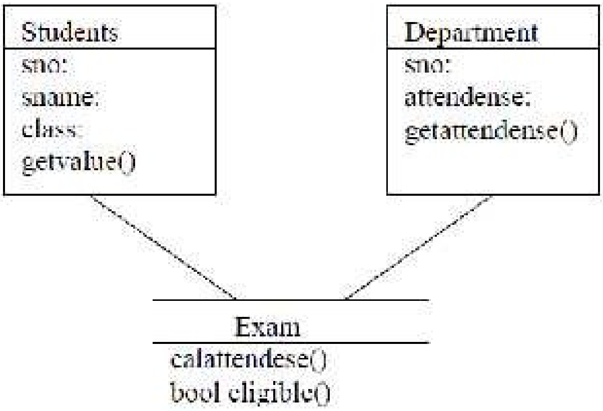
1. Write a java program to calculate gross salary & net salary taking the following data.
   Input : empno,empname,basic
   Process: DA=50%of basic
   HRA=25%of basic
   PF=10%of basic
   PT=Rs100/-

2. Write a java program that implements educational hierarchy using inheritance.



3. Write a program to identify the accessibility of a variable by means of different access specifies within and outside package.

4. Write a java program to find the details of the students eligible to enroll for the examination (Students, Department combined give the eligibility criteria for the enrollment class) using interfaces



5. Write a Java program that displays area of different Figures (Rectangle, Square, Triangle) using the method overloading.

6. Write a Java program that displays that displays the time in different formats in the form of HH,MM,SS using constructor Overloading. .
7. Write a Java program that counts the number of objects created by using static variable.
8. Write a Java program to count the frequency of words, characters in the given line of text.
9. Write a Java program for sorting a given list of names in ascending order.
10. Write a Java program that reads a line of integers separated by commas and then displays each integer and fund the sum of he integers (using String Tokenizer)
11. Write a Java program that reads a file name from the user then displays information about whether that file exists, file is writable, the type of file and length of the file in bytes.
12. Write a Java program that reads a file and displays the file on the screen with a line number before each line.
13. Write a Java program that reads a file and displays the no of lines and words in that file.
14. Write a Java program that reads to copy source seat. File and display on the console
15. Write a java program that implements Array Index out of bound Exception using built-in-Exception.
16. Write a java program that implements bank transactions using user denied exception.
17. Write a java program to identify the significance of finally block in handling exceptions.
18. Write a java program to generate multiple threads of creating clock pulses.(using runnable interface)
19. Write a java program to identify the use of synchronized blocks to synchronized methods.
20. Write an applet to display a simple message on a colored background.
21. Write an applet to display a moving banner showing the status of it.
22. Write an applet to draw a simple and beautiful landscape.
23. Write a java program to demonstrate key events by using Delegation event model.
24. Write a java program to implement mouse events like mouse pressed, mouse released and mouse moved by means of adapter classes.
25. Write a java program to demonstrate window events on frame.
26. Write an applet that computes the payment of a loan based on the amount of he loan, interest rate and the number of months.
27. Write an applet to perform the 4 basic arithmetic operations as buttons in a form accepting two integers in textboxes and display their result.
28. Write a java program to design a registration form for creating a new eMail account.
29. Write a java program to design the page authenticating user name and password by using SWING.
30. Write a java program to design a calculator by using Grid Layout.

Reference Books:

1. Herbet Schidt and Dale Srien, Java Fundamentals - A comprehensive Introduction, TMH.
2. P.J. Deitel and H.M. Deitel, Java for Programmers, Pearson education
3. P.J. Deitel and H.M. Deitel, Java: How to Program, PHI.

Quantitative & Verbal Aptitude-I
(Common for all Branches)

IT319 **Credits: 2**

Instruction: 4 Periods/week Sessional Marks: 100

Prerequisite:

Course Objectives:

Quantitative Aptitude–I

- To prepare the students on various principles related to numerical computations.
- To explain concepts related to numerical estimation.
- To illustrate and explain the fundamentals related to geometry and mensuration.

Verbal Aptitude–I

- To categorize and explain principles of grammar in order to minimize errors in English.
- To list and quote high frequency words by giving relevant examples.
- To categorize, apply and use data as per the requirement.
- To construct and make use of idioms, phrasal verbs and other expressions used in professional contexts.
- To critically evaluate reading material for better comprehension

Course Outcomes:

Quantitative Aptitude–I

After completion of this course, a student will be able to:	
1.	Solve problems related to numerical computations in company specific and other competitive tests.
2.	Able to recall and use the concepts to solve problems numerical estimation with respect to company specific and competitive tests.
3.	Apply basic principles related to geometry and mensuration & solve questions in company specific and competitive tests.

Verbal Aptitude–I

After completion of this course, a student will be able to:	
1.	Detect grammatical errors in the text/sentences and rectify them while answering their competitive / company specific tests and frame grammatically correct sentences while writing.
2.	Answer questions on synonyms, antonyms, hyponyms, hypernyms and other vocabulary based exercises while attempting company specific and other competitive tests.
3.	Use their logical thinking ability and solve questions related to reasoning based exercises.
4.	Choose the appropriate word/s/phrases suitable to the given context in order to make the sentence/paragraph coherent.
5.	Analyze the given data/text and find out the correct responses to the questions asked based on the reading exercises; identify relationships or patterns within groups of words or sentences.

Mapping of course outcomes with program outcomes:

CO		PO												PSO	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
QA	1														
	2														
	3														
VA	4														
	5														
	6														
	7														
	8														

SYLLABUS

Section –A (Quantitative Aptitude –I)

UNIT I **6 Periods**
Numerical computation:
Applications based on Numbers, Chain Rule, Ratio Proportion

UNIT II **6 Periods**
Numerical estimation - I
Applications Based on Time and work, Time and Distance

UNIT III **4 Periods**
Numerical estimation – II
Applications based on Percentages, Profit Loss and Discount, Simple interest and Compound Interest

Partnerships, Shares and dividends

UNIT IV **4 Periods**
Data interpretation
Data interpretation related to Averages, Mixtures and allegations, Bar charts, Pie charts, Venn diagrams

UNIT V **4 Periods**
Application to industry in Geometry and Mensuration

Books for practice

1. Quantitative aptitude by RS Agarwal, S Chand Publications
2. Verbal and non verbal Reasoning by RS Agarwal from S Chand publications

References

1. Barron’s by Sharon Welner Green and Ira K Wolf (Galgotia Publications pvt. Ltd.)
2. Quantitative Aptitude by U Mohan Rao Scitech publications
3. Quantitative Aptitude by Arun Sharma McGrawhill publications
4. Quantitative Aptitude by Ananta Asisha Arihant publications
5. Quantitative Aptitude by Abhijit Guha
6. Quantitative Aptitude by Pearson publications
7. Material from ‘IMS, Career Launcher and Time’ institutes for competitive exams.
8. Elementary and Higher algebra by HS Hall and SR knight.

Websites:

1. www.m4maths.com
2. www.Indiabix.com
3. 800score
4. Official GRE site
5. Official GMAT site

Section –B (Verbal Aptitude –I)

UNIT I **7 Periods**

Grammar:

Parts of speech(with emphasis on appropriate prepositions, co-relative conjunctions, pronouns-number and person, relative pronouns), articles(nuances while using definite and indefinite articles), tenses(with emphasis on appropriate usage according to the situation), subject – verb agreement (to differentiate between number and person), clauses (use of the appropriate clause, conditional clauses), phrases(use of the phrases, phrasal verbs), degrees of comparison (comparing apples and oranges, comparison and number), modifiers(misplaced and dangling modifiers, absence of modifiers), determiners, parallelism in structure(symmetry in two part sentences), word order, subjunctive mood, redundancy, special types of sentences, miscellaneous types, identifying errors in a given sentence, correcting errors in sentences.

UNIT II **4 Periods**

Vocabulary:

Synonyms and synonym variants (with emphasis on high frequency words), antonyms and antonym variants (with emphasis on high frequency words), homonyms, hyponyms, hypernyms and General idioms.

UNIT III **5 Periods**

Reasoning:

Critical reasoning (understanding the terminology used in CR- premise, assumption, inference, conclusion), Sequencing of sentences (to form a coherent paragraph, to construct a meaningful and grammatically correct sentence using the jumbled text), to use logical reasoning and eliminate the unrelated word from a group.

UNIT IV **4 Periods**

Usage:

Sentence completion (with emphasis on signpost words and structure of a sentence), contextual meanings (to use the appropriate word according to the situation), supplying a suitable beginning/ending/middle sentence to make the paragraph coherent, idiomatic language (with emphasis on business communication), punctuation depending on the meaning of the sentence, run on errors, sentence fragments, coma splices.

UNIT V **4 Periods**

Reading Comprehension:

Types of passages (to understand the nature of the passage), types of questions (with emphasis on inferential and analytical questions), style and tone (to comprehend the author's intention of writing a passage), strategies for quick and active reading(importance given to skimming, scanning), summarizing,reading between the lines, reading beyond the lines, techniques for answering questions related to vocabulary (with emphasis on the context), supplying suitable titles to the passage, identifying the theme and central idea of the given passages.

Books for Practice

1. A. J. Thomson, A. V. Martinet, Practical English Grammar by Oxford University press
2. FT wood, Remedial English Grammar for Foreign Students published by Macmillan Publishers
3. Edgar Torpe, Showick Thorpe, Objective English by Pearson Education
4. Cambridge and Oxford Dictionaries

Reference Books and websites:

1. Barron's by Sharon Welner Green and Ira K Wolf (Galgotia Publications Pvt.Ltd.)
2. Websites: Indiabix, 800 score, official CAT, GRE and GMAT sites
3. Material from 'IMS, Career Launcher and Time' in`stitutes for competitive exams.
4. Collins Cobuild English Grammar by Goyal Publishers
5. Word Power Made Easy by Norman Lewis-Goyal Publishers

SEMESTER-2

COMPILER DESIGN

IT321 **Credits: 4**

Instruction: 4 Periods & 1 Tut /Week Sessional Marks: 40

End Exam: 3 Hours End Exam Marks: 60

Prerequisite:

Formal Languages and Automata Theory

Course Objectives:

- Introduce the major concept areas of language translation and compiler design.
- Develop an awareness of the function and complexity of modern compilers.
- Provide practical, hands-on experience in compiler design, writing and modification.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Summarize finite automata, regular expressions and analyze compiler architecture.
2.	Analyze lexical analyzer and construct parsers
3.	Analyze symbol table and generate Intermediate code.
4.	Analyze register allocation and Synthesize the code optimization techniques.

Mapping of course outcomes with program outcomes:

	PO												PSO	
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	2	1	3	2	0	1	0	1	1	1	0	2	2	2
CO2	2	3	3	2	2	1	0	1	1	1	0	2	2	2
CO3	3	2	1	1	1	1	0	1	1	1	0	2	2	2
CO4	2	2	3	2	2	1	0	1	1	1	0	2	2	2

SYLLABUS

UNIT I: **10 Periods**

Overview of language processing: preprocessors, compiler, assembler, interpreters, Linkers & loaders, Basic Concepts of Finite State Systems, Deterministic and Non-Deterministic Finite Automata, Regular expressions, Regular sets and Regular Grammars.

Overall view of Compilers: Brief discussion on various phases of Compilers.

UNIT II: **10 Periods**

Design of lexical analyzer: Lexical Analysis, Role of Lexical Analysis, Lexical Analysis Vs.Parsing, Token, patterns and Lexemes, Lexical Errors, Regular Expressions, Regular definitions for the language Constructs, Strings, Sequences, Comments, Transition diagram for recognition of tokens, Reserved words and identifiers, Examples.

UNIT III: **14 Periods**

Syntax Analysis: Role of a parser, classification of parsing techniques, Top down parsing- First and Follow, LL 1 Grammars, Non-Recursive predictive parsing, Error recovery in Predictive Parsing. Bottom Up parsing- Operator Precedence, Shift Reduce Parsing, Difference between LR and LL Parsers, Construction of SLR parsing tables, CLR parsing tables, LALR parsing tables.

UNIT IV: **14 Periods**

Semantic analysis: SDT, Postfix notation, parsing tree, Intermediate code, Three address Code, Quadruples, Triples , symbol tables, use of symbol tables. Runtime Environment: storage organization, stack allocation, access to non-local data, heap management, Parameter passing mechanisms

UNIT V: **12 Periods**

Intermediate Code Optimization: The principle sources of optimization, Loop Optimization, DAG, Global data flow analysis.

Code Generation: Problems, Machine model, A simple code generator, Register allocation and assignment, Code generation from DAG, Peep hole optimization

Text Books:

1. Aho, D. Ullman, Principles of Compiler Design

Reference Books:

1. Kenneth. C. Louden, Compiler Construction, Vikas Pub. House.

DESIGN AND ANALYSIS OF ALGORITHMS

IT322 **Credits: 4**
Instruction: 4 Periods & 1 Tut /week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:
C Programming, Data Structures

Course Objectives:

- Assess how the choice of data structures impacts the performance of programs.
- Choose the appropriate data structure and algorithm design method for specified problems.
- Evaluate and Analyze the time complexities for various problems

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyze the time complexity of recursive and non-recursive algorithms with respect to the asymptotic order of growth
2.	Design and analyze algorithms to solve optimization problems using Divide and Conquer and Decrease and Conquer.
3.	Design and analyze algorithms to solve optimization problems using Transform and Conquer, Space and Time Tradeoffs, Dynamic Programming and Greedy Techniques.
4.	Reduce one NP-complete problem to another NP-complete problem in polynomial time and analyze polynomial time heuristics to approximate solutions for NP complete problems.

Mapping of course outcomes with program outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3						1	1	1		1	3	2
CO2	2	3	3	1				1	1	1		1	3	2
CO3	2	3	3	1				1	1	1		1	3	2
CO4	2	3		1				1	1	1		1	3	2

SYLLABUS

UNIT I: **16 Periods**

Introduction: Fundamentals of algorithmic problem solving – important problem types – fundamental data structures.

Fundamentals of analysis of algorithms and efficiency: Analysis framework – Asymptotic Notations and Basic Efficiency classes – Mathematical Analysis of Non-recursive Algorithms – Mathematical Analysis of recursive Algorithms.

Brute Force: Selection Sort and Bubble sort – Sequential Search and Brute – Force String Matching – Closest Pair and Convex-Hull Problems by Brute Force – Exhaustive Search.

UNIT II: **12 Periods**

Divide-and-Conquer: Mergesort – Quicksort –Binary Search –Binary Tree Traversals and Related Properties – Multiplication of large integers and Strassen's Matrix Multiplication – Closest-Pair Convex-Hull Problems by Divide-and –Conquer.

Decrease – and – Conquer: Insertion Sort –Depth-First Search and Breadth-First Search - Topological Sorting –Algorithms for Generating Combinatorial Objects – Decrease – by – a – Constant - Factor Algorithms – Variable – Size – Decrease Algorithms.

UNIT III: **12 Periods**

Transform-and-Conquer: Presorting – Gaussian Elimination – Balanced Search Trees – Heaps and Heapsort – Horner's Rule and Binary Exponentiation – Problem Reduction.

Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in string Matching – Horspool's algorithm, Hashing, B-Trees.

UNIT IV: **12 Periods**

Dynamic Programming: Computing a Binomial Coefficient –Warshall's and Floyd's Algorithm–Optimal Binary Search Trees -The Knapsack Problem and Memory Functions.

Greedy Technique: Prim's Algorithm –Kruskal's Algorithm –Dijkstra's Algorithm –Huffman Trees.

UNIT V: **12 Periods**

Limitations of Algorithm Power: Lower-Bound Arguments –Decision Trees –P, NP, NP hard and NP –complete problems**.**

Coping with the Limitations of Algorithms Power: Backtracking-n Queens, Hamiltonian circuit, subset sum problem, –Branch-and-Bound-Assignment, Knapsack, Traveling Salesman problems.

Text Books

1. Anany Levitin, Introduction to Design & Analysis of Algorithms, Pearson Education, NewDelhi, 2003

Reference Books:

1. Thomas H. Corman, Charles E. Leiserson, Ronald R. Rivest & Clifford Stein,Introduction to Algorithms, Prentice Hall of India, New Delhi, New Delhi
2. Aho, Hopcroft & Ullman ,The Design and Analysis of computer Algorithms, PearsonEducation, New Delhi, 2003
3. Gilles Brassard & Paul Bratley,Fundamentals of algorithmics, Prentice Hall of India, New Delhi

OBJECT ORIENTED ANALYSIS AND DESIGN WITH UML

IT323 **Credits: 3**

Instruction: 3 Periods & 1 Tut /Week Sessional Marks: 40

End Exam: 3 Hours End Exam Marks: 60

Prerequisite:

Object oriented concepts, C++ programming, Fundamentals of Software Engineering

Course Objectives:

- Learn the basics of OO analysis and design skills.
- Learn the UML design diagrams.
- Learn to map design to code.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyze the problem using object oriented concepts like Class, Object, Inheritance, Encapsulation, Abstraction and Polymorphism.
2.	Design class model for the Structure of an object, State model for time and sequence operations, Interaction model to tell the behavior of the system of given project.
3.	Design Interaction modeling by using Use case model, Sequence model and activity model of given project.
4.	Synthesize the application analysis using application Interaction, Class model, state model of a given project.

Mapping of course outcomes with program outcomes:

		PO												PSO	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	2	2	3	2	3				2	2	2	2	2	2
	2	2	2	3	2	3				2	2	2	2	2	3
	3	2	3	3	2	3				2	2	2	3	2	3
	4	3	3	3	2	3				2	2	2	3	2	3

SYLLABUS

UNIT I: **8 Periods**

Introduction: Object orientation and development, Themes, Evidences for usefulness of OO Development, Modeling History

Modeling as a Design Technique: Modeling, Abstraction, The three models.

Class modeling: Object and class concepts, links and association concepts, generalization and inheritance, a sample class model, navigation of class models

UNIT II: **10 Periods**

Advanced class modeling: Advanced object and class concepts, association ends,N-ary associations,Aggregation, Abstract classes, multiple inheritance, Metadata, Reification, Constraints,derived Data, Packages

State modeling: Events, States, Transitions and conditions,State diagrams, State diagram Behavior.

Advanced state modeling: Nested state diagrams, nested states, signal generalization, concurrency, a sample state model, relation of class and state models

UNIT III: **10 Periods**

Interaction Modeling: Use Case Model, Sequence models, Activity models

Advanced Interaction Modeling: Use case relationships, Procedure sequence models, special constructs for Activity Models

Concepts Summary: class Model, State Model, Interaction Model, Relationship among the models

UNIT IV: **10 Periods**

Process overview: Development stages, Development Life cycle

System conception: Devising a system concepts, elaborating a concept, preparing a problem statement.

Domain Analysis: Overview, domain class model, domain state model, domain interaction model, iterating the analysis

UNIT V: **10 Periods**

Application Analysis: Application Interaction Model, Application class model, Application state model, adding operations

System Design: overview, estimating performance, making a reuse plan, breaking a system into subsystems, identifying concurrency, allocation of subsystems, management of data storage, handling Global resources, choosing a software control strategy, handling boundary conditions, setting trade-off priorities Common architectural styles, architecture of ATM system

Class Design: Overview, bridging a gap, realizing use case, designing algorithms, Recursing downward, refactoring, design optimization, reification of behavior, Adjustment of inheritance, organizing a class design, ATM Example

Text Books:

1. Michael Blaha and James Rumbaugh, Object Oriented Modeling and Design with UML, Prentice Hall India, 2nd Edition.

Reference Books:

1. Grady Booch, Object Oriented Analysis and Design with Applications, Pearson Education Asia
2. Berno Bruegge, Allen H. Dutoit,,Object Oriented Software Engineering - Pearson Education Asia
3. H. Srimathi, H. Sriram, A. Krishnamoorthy, Object Oriented Analysis and Design using UML

MOBILE COMPUTING AND APPLICATION DEVELOPMENT

IT324 **Credits: 3**

Instruction: 3 Periods & 1 Tut /Week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:

Object oriented concepts, Java Programming, Database Concepts

Course Objectives:

- Understand the basic concepts App Development
- Be familiar with Android system and user interface

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyze the fundamentals of Android Operating System & Application Program Skills
2.	Design Layouts, Views and User Interfaces on Android Development IDEs
3.	Apply Database concepts for Android Applications and Interfaces
4.	Develop and publish Mobile applications on Android devices

Mapping of course outcomes with program outcomes:

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	PSO1	PSO2
CO1	2	2	3	3	3				3		3	2	2	2
CO2	2	3	3	3	3				3		3	2	2	3
CO3	3	3	3	3	3				3		3	3	2	3
CO4	2	3	3	3	3				3		3	3	2	3

SYLLABUS

UNIT I: **8 Periods**

Introduction To Mobile Apps and Android: Need of Mobile Apps, Different Kinds of Mobile Apps, Android History.

Android Architecture: Overview of Android Stack, Android Features, Introduction to OS layers

Deep Overview in Android Stack: Linux Kernel, Libraries, Android Runtime, Application Framework, Dalvik VM

UNIT II: **10 Periods**

Installing Android Machine: Configuring Android Stack, Setting up Android Studio, Working with Android Studio, Using Older Android Tools

Creating First Android Application: Creating Android Project, Debugging Application setting up environment, AVD Creation, Executing Project on Android Screen

Android Components: Activities, Services, Broadcast Receivers, Content Providers

Hello World App: Creating your first project, The manifest file, Layout resource, Running your app on Emulator

UNIT III: **10 Periods**

Building UI with Activities: Activities, Views, layouts and Common UI components, Creating UI through code and XML, Activity lifecycle, Intents, Communicating data among Activities

Advanced UI: Selection components (GridView, ListView, Spinner), Adapters, Custom Adapters, Complex UI components, Building UI for performance, Menus, Creating custom and compound Views

Notifications: Toast, Custom Toast, Dialogs, Status bar Notifications

Multithreading: Using Java Multithreading classes, AsyncTask, Handler, Post, Writing an animated game

UNIT IV: **10 Periods**

Styles And Themes: Creating and Applying simple Style, Inheriting built-in Style and User defined style, Using Styles as themes

Resources and Assets: Android Resource, Using resources in XML and code, Localization Handling Runtime configuration changes

Intent, Intent Filters and Broadcast Receivers: Role of filters, Intent-matching rules, Filters in your manifest, Filters in dynamic Broadcast Receivers, Creating Broadcast receiver

Receiving System Broadcast: Understanding Broadcast action, category and data, Registering Broadcast receiver through code and through XML, Sending Broadcast

UNIT V: **10 Periods**

Data Storage: Shared Preferences, Android File System, Internal storage, External storage, SQLite, IntroducingSQLite, SQLiteOpenHelper and creating a database, Opening and closing adatabase, Working with cursors Inserts, updates, and deletes .

Content Providers: Accessing built in content providers, Content provider MIME types Searching for content, Adding, changing, and removing content, Creating content provider Working with content files

Multimedia in Android: Multimedia Supported audio formats, Simple media playback, Supported video formats, Simple video playback, Android Application Deployment, Introduction to xamarin

Text Books:

1. Pradeep kothari,Android Application Development (With Kitkat Support), Black Book, Dreamtech publications

Reference Books:

1. Prasant-Pattnaik, Fundamentals of Mobile Computing
2. Padmini, Android App Development: A Complete Tutorial For Beginners
3. Reto Meier, Professional-Android-Application-Development, Wrox Publications

DISTRIBUTED OPERATING SYSTEMS

IT325 (A) **Credits: 4**
Instruction: 4 Periods & 1 Tut /week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:
Operating Systems, Computer Networks

Course Objectives:

- Understand foundations of Distributed Systems.
- Introduce the idea of peer to peer services and file system.
- Understand in detail the system level and support required for distributed system.
- Understand the issues involved in studying process and resource management.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Illustrate the fundamentals of distributed systems and networks to design a distributed system
2.	Apply message passing and RPC concepts to provide communication in Distributed Systems
3.	Analyze the implementation of distributed Shared Memory and Synchronization in Distributed Systems
4.	Apply the knowledge of Process and Resource Management on distributed systems
5	Summarize distributed File System and Naming concepts and apply to distributed systems related applications

Mapping of course outcomes with program outcomes:

	PO1 App	PO2 anal	PO3 des	PO4 inv	PO5 tools	PO6 engg	PO7 prof	PO8 ethics	PO9 eam	PO10 comm	PO11 econ	PO12 LL	PSO1	PSO2
O-1	3	1				1	1		1	1	1	2	3	2
O-2	3	2		1		1	1		1	1	1	2	3	2
O-3	2	3	1	2		1	1	1	1	1	1	2	3	2
O-4	3	2	1	1		1	1	1	1	1	1	2	3	2
O-5	2	2	3	2		1	1	1	1	1	1	2	3	2

SYLLABUS

UNIT I: Fundamentals and networks **10 periods**
Fundamentals: Distributed computing system, evolution, models, popularity, Distributed operating system, design issues, introduction to DCE;
Computer networks: Introduction, Types, LAN, WAN, Communication protocols, Internetworking.

UNIT II: Message passing and RPC **14 periods**
Message passing: Introduction, features, issues in IPC, synchronization, Buffering, multi datagram messages, encoding and decoding, process addressing, failure handling, group communication.

Remote procedure call: Introduction, RPC model, Transparency of RPC, Implementation, Stub generation, RPC messages, server management, parameter-passing semantics, call semantics, communication protocols, complicated RPC's, Client-Server Binding, exception handling, security, some special types of RPCs, RPC in heterogeneous environments, lightweight RPC.

UNIT III: Distributed shared memory and synchronization 12 periods
Distributed Shared memory: Introduction, general architecture, design and implementation issues, granularity, structure of shared memory space, consistency models, replacement strategy, thrashing, other approaches to DSM, Heterogeneous DSM, Advantages. **Synchronization:** Introduction, Clock Synchronization, Event ordering, Mutual Exclusion, Deadlock, Election Algorithms.

UNIT IV: Resource management and process management 10 periods
Resource management: Introduction, Desirable Features of a good global scheduling algorithm, Task assignment approach, load-balancing approach, load-sharing approach. **Process Management**: Introduction, process migration, Threads

UNIT V: Distributed file systems and naming 12 periods
Distributed file systems: Introduction, features, file models, Accessing models, sharing models, file- caching schemes, file Replication, Fault tolerance, Atomic transactions, design principles. **Naming**: Introduction, features, fundamental terminologies, system-oriented names, object-locating mechanisms, human-oriented names, name caches, naming and security.

Text Books:

1. pradeep k. Sinha, “Distributed Operating Systems; concepts and design”, Edition, Pearson Education,

Reference Books:

1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.
2. Tanenbaum A.S., Van Steen M., “Distributed Systems: Principles and Paradigms”, Pearson Education, 2007.
3. Liu M.L., “Distributed Computing, Principles and Applications”, Pearson Education, 2004.
4. Nancy A Lynch, “Distributed Algorithms”, Morgan Kaufman Publishers, USA, 2003.

TECHNOLOGY MANAGEMENT

IT325 (B) **Credits: 4**
Instruction: 4 Periods & 1 Tut /week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:
Fundamentals of computers and Databases

Course Objective: To understand about Information Technology and Information systems, Functional Information systems, DBMS, Disaster Management, Data Mining and Cloud Computing.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyze Information systems – Decision Support system, Knowledge Management System, Executive support system, Geographic information system and International Information System.
2.	Analyze data ware house concepts and software development methodologies.
3.	Assess latest business initiatives such as E-Business, E-Governance and cloud computing emerging in the field of information technology.
4.	Determine the concepts related to network security such as error detection, Disaster management and computer crimes.

Mapping of course outcomes with program outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3						1	1	1	2	1	2	1
CO2	3	3	3	3	3			1	1	1	2	2	2	2
CO3				2				1	1	1	2	2	2	2
CO4	3	2	2	2	2			1	1	1	2	2	2	2

SYLLABUS

UNIT I: **12 Periods**
Introduction - Data, Information, Intelligence, Information Technology, Information System, evolution, types based on functions and hierarchy, Functional Information Systems, DSS, EIS, KMS, GIS, International Information System.

UNIT II: **14 Periods**
Systems analysis and design - Systems development methodologies, Systems Analysis and Design Tools – System flow chart, Decision table, DFD, ER, Object oriented Analysis and Design, UML diagram.

UNIT III: **14 Periods**
Database management systems - DBMS – HDBMS, NDBMS, RDBMS, OODBMS, Query Processing, SQL, Concurrency Management, Data warehousing and Data Mart

UNIT IV: **12 Periods**

Security, control and reporting - Security, Testing, Error detection, Controls, IS Vulnerability, Disaster Management, Computer Crimes, Securing the Web, Intranets and Wireless Networks, Software Audit, Ethics in IT, User Interface and reporting.

UNIT V: **12 Periods**

New it initiatives - Role of information management in ERP, e-business, egovernance, Data Mining, Business Intelligence, Pervasive Computing, Cloud computing, CMM.

Text Books:

1. Robert Schultheis and Mary Summer, Management Information Systems – The Managers View, Tata McGraw Hill, 2008.

Reference Books:

1. Gordon Davis, Management Information System: Conceptual Foundations, Structure and Development, Tata McGraw Hill, 7th edition, 2006.
2. Haag, Cummings and Mc Cubbrey, Management Information Systems for the Information Age,McGraw Hill, 2012.
3. Turban, McLean and Wetherbe, Information Technology for Management – Transforming Organisations in the Digital Economy, John Wiley, 6th edition, 2009.
4. Raymond McLeod and Jr. George P. Schell, Management Information Systems, Pearson Education, 2007.
5. James O Brien, Management Information Systems – Managing Information Technology in the E-business enterprise, Tata McGraw Hill, 2010.
6. Corey Schou and Dan Shoemaker, Information Assurance for the Enterprise – A Roadmap to Information Security, Tata McGraw Hill, 2007.
7. Frederick Gallegor, Sandra Senft, Daniel P. Manson and Carol Gonzales, Information Technology Control and Audit, Auerbach Publications, 4th edition, 2012.

SOFTWARE TESTING AND AUTOMATION

IT325 (C) **Credits: 4**
Instruction: 4 Periods & 1 Tut /week Sessional Marks: 40
End Exam: 3 Hours End Exam Marks: 60

Prerequisite:
Any programming language, Object oriented concepts, Fundamentals of Software Engineering

Course Objectives:
The objective is to teach the students about Software testing, it helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make it sure that it's performing well and as per the specifications.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Apply software testing knowledge and engineering methods using modern software development
2.	Design Test-cases using testing strategies
3.	Manage incidents and risks within a project using testing and debugging policies
4.	Develop a test tool to support test automation

Mapping of course outcomes with program outcomes:

		PO												PSO	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	2	3	2	1						3	2	2	3	2
	2	2	3	2	1						3	2	2	3	2
	3	3	2	2	2	2					2	2	3	2	2
	4	2	2	2	2	1					2	3	2	2	2

SYLLABUS

UNIT I: **12 Periods**

Introduction: Testing as an Engineering Activity – Role of Process in Software Quality – Testing as a Process – Basic Definitions – Software Testing Principles – The Tester's Role in a Software Development Organization –Origins of Defects – Defect Classes – The Defect Repository and Test Design – Defect Examples –Developer/Tester Support for Developing a Defect Repository.

UNIT II: **13 Periods**

Test case design: Introduction to Testing Design Strategies – The Smarter Tester – Test Case Design Strategies – Using Black Box Approach to Test Case Design Random Testing – Requirements based testing – positive and negative testing – Boundary Value Analysis –

decision tables - Equivalence Class Partitioning state based testing – cause effect graphing – error guessing - compatibility testing – user documentation testing – domain testing - Using White–Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing - Coverage and Control Flow Graphs – Covering Code Logic – Paths – Their Role in White–box Based Test Design – code complexity testing – Evaluating Test Adequacy Criteria.

UNIT III: **12 Periods**

Levels of Testing: The Need for Levels of Testing – Unit Test – Unit Test Planning –Designing the Unit Tests- The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests– Integration Test Planning – scenario testing –defect bash elimination -System Testing – types of system testing - Acceptance testing –performance testing - Regression Testing – internationalization testing – Adhoc testing -Alpha – Beta Tests – testing OO systems – usability and accessibility testing

UNIT IV: **12 Periods**

Test management : People and organizational issues in testing – organization structures for testing teams – testing services -Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management– test process - Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group.

UNIT-V: **12 Periods**

Controlling and monitoring: Software test automation – skills needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation - Test metrics and measurements –project, progress and productivity metrics – Status Meetings – Reports and Control Issues – Criteria for Test Completion – SCM – Types of reviews – Developing a review program – Components of Review Plans– Reporting Review Results. – Evaluating software quality – defect prevention – testing maturity model

Text books:

1. Srinivasan Desikan and Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson education, 2006.
2. Ilene Burnstein, “Practical Software Testing”, Springer International Edition, Chennai, 2003

Reference books:

1. Boris Beizer, “Software Testing Techniques”, Second Edition, Dreamtech,
2. Elfriede Dustin, “Effective Software Testing”, First Edition, Pearson Education, Renu Rajani, Pradeep Oak, “Software Testing – Effective Methods, Tools and Techniques”, Tata McGraw Hill, 2004.

CLIENT SERVER TECHNOLOGIES

IT325 (D)	**Credits: 4**
Instruction: 4 Periods & 1 Tut /week	Sessional Marks: 40
End Exam: 3 Hours	End Exam Marks: 60

Prerequisite:

Computer Networks, Operating Systems

Course Objectives:

- Students will be able to provide a generalized definition of client-server computing, one not limited to simple distributed database systems.
- Students will learn the advantages of client-server systems over monolithic systems.
- Students will provide definitions and explanations for a large number of technical terms and acronyms related to client-server computing.
- Students will apply the techniques and features of a client/server development language to construct a moderately complex client/server application.
- Students will apply the concepts learned in this course to the development of client-server applications that are Internet and/or World Wide Web based.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Introduce the client server architecture and fundamentals of distributed systems
2.	Understand applications and computing aspects of client/ server approach
3.	Analyze the hardware and software structure of a client/server model
4.	Understand the categories, classes and environment of server
5	Understand the structural design of server operating systems

Mapping of course outcomes with program outcomes:

		PO												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	1	2	3	2	1						3	2	2	1
	2	1	2	3	2	1						3	2	2	2
	3	3	3	2	2	2	2					2	2	2	3
	4	2	2	2	2	2	1					2	3	2	3
		3	3	3	2	2	2					2	2	2	3

SYLLABUS

UNIT I: **12 Periods**

Introduction to client/server computing-What is client/server computing-Benefits of client/server computing-Evolution of C/S computing-Hardware trends-Software trends-Evolution of operating systems-Networking N/W trends-Business considerations.

UNIT II: **12 Periods**

Overview of C/S Applications: Components of C/S Applications-Classes of C/S Applications-Categories of C/S Applications. Understanding C/S Computing: Dispelling of myths - Obstacles-Upfront &Hidden-Open systems & Standards-Standard-Setting Organizations-Factors for success.

UNIT III: **15 Periods**

The client hardware and software: Client Component-Client operating Systems-what is GUI-Database Access-Client Software Products: GUI Environments –Converting 3270/5250 Screens-Database Tools-Client Requirements-GUI Design Standards-Interface Independence-Testing Interfaces.

UNIT IV: **12 Periods**

The Server: Categories of Servers –Features of Server Machines-Classes of Server machines-Server Environment: N/W Management Environment-N/W computing Environments-Extensions-Network Operating System-Loadable Module.

UNIT V: **13 Periods**

Server operating system: OS/2. Windows new technology-Unix based OS-Server requirements: Platform independence-Transaction processing-Connectivity-Intelligent database-Stored procedure-Triggers-Load leveling-Optimizer-Testing and diagnostic tools-Backup recovery mechanisms.

Text Book:

1. Dawna Travis Dewire-“Client/Server Computing”-Tata McGraw Hill, 1994

Reference Books:

1. Umar, Amjad-“Object-Oriented Client/Server Internet Environments”- Prentice Hall
2. *Object-Oriented Client/Server Internet Environments*, Amjad Umar, Prentice Hall PTR, Upper Saddle River, New Jersey, 1997

MOBILE COMPUTING AND APPLICATION DEVELOPMENT LAB

IT326 **Credits: 2**

Practical: 3 Practical periods /Week Sessional Marks: 50

End Exam: 3 Hours End Exam Marks: 50

Prerequisite:

Object oriented concepts, Java Programming, Database Concepts

Course Objectives:

- Understand the basic concepts App Development
- Be familiar with Android system and user interface

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyze the Basic Concepts and Techniques for Creating and DesigningAndroid Applications
2.	Develop and Publish the Android Applications on the Android Devices

Mapping of course outcomes with program outcomes:

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	PSO1	PSO2
CO1	2	2	3	3	3				3		3	2	2	2
CO2	3	3	3	3	3				3		3	3	2	3

Requirements:

Standalone desktops with Windows or Android or iOS or Equivalent Mobile Application Development Tools with appropriate emulators and debuggers

List of Experiments:

1. Installation of mobile application development tool and sample programs
2. Develop an application that uses GUI components, Font and Colors
3. Develop an application that uses Layout Managers and event listeners.
4. Develop a native calculator application.
5. Write an application that draws basic graphical primitives on the screen.
6. Develop an application that makes use of database.
7. Develop an application that makes use of RSS Feed.
8. Implement an application that implements Multi threading
9. Develop a native application that uses GPS location information.
10. Implement an application that writes data to the SD card.
11. Implement an application that creates an alert upon receiving a message.
12. Write a mobile application that creates alarm clock

Reference Books:

1. Pradeep kothari,Android Application Development (With Kitkat Support), Black Book, Dreamtech publications
2. Prasant-Pattnaik,Fundamentals of Mobile Computing
3. Padmini,Android App Development: A Complete Tutorial For Beginners
4. Reto Meier,Professional-Android-Application-Development,Wrox Publications

Web References:

1. https://developer.android.com/training/basics/firstapp/creating-project.html
2. https://www.tutorialspoint.com/android/android_studio.htm
3. https://www.linuxhelp.com/how-to-install-android-studio-in-ubuntu/

WEB BASED OPEN SOURCE TECHNOLOGIES LAB

IT327 **Credits: 3**
Practical: 3 Periods & 1 Tut /Week Sessional Marks: 50
End Exam: 3 Hours End Exam Marks: 50

Prerequisite:
Object oriented concepts, any programming language.

Course objectives:

- Students will gain the skills and project-based experience needed for entry into web design and development careers.
- Students will be able to use a variety of strategies and tools to create websites.
- Students will develop awareness and appreciation of the myriad ways that people access the web and will be able to create standards-based websites that are accessible and usable by a full spectrum of users.

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Analyze the creation of web pages using HTML\XHTML\CSS.
2.	Discover different types of images used in modern webpage along with different functions
3.	Demonstrate web development platforms with display resolutions, viewports & browsers that render websites
4.	Design web pages with development frameworks like Drupal, Bootstrap, Django, web services & Content Management Systems like Joomla
5.	Develop web based application using suitable client side & server side web technologies like Content Management Systems& web services. dents will be able to

Mapping of course outcomes with program outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	1	3				2		2	2	1	3
CO2	3	2	1	1	3				2		2	2	2	2
CO3	3	3	1	2	3				2		2	2	2	3
CO4	3	1	1	2	3				3		3	3	1	3
CO5	3	3	1	2	3				3		3	2	3	3

LIST OF THE EXPERIMENTS TO BE DONE ON THE FOLLOWING TOPICS

1. Introduction
2. HTML5
3. CSS
4. PHP
5. Python
6. Content Management Systems
7. Joomla
8. Drupal
9. Bootstrap
10. MySql
11. Different types of servers (XAMP, WAMP)

LIST OF THE EXPERIMENTS.

1. Installation of frame works in different operating systems.
2. Design Registration form using HTML5.
3. Design Login Form using HTML5.
4. Design and Develop Responsive Website using PHP and MySQl.
5. Design and Develop Responsive Website using Content Management Systems.

* Each application requires 2 to 3 weeks to finish.

Reference Books:

1. HTML 5 Black book, 2nd edition, Dream tech press.
2. David Sklar, PHP cock book, O'Reilly media, 3rd edition.
3. Luke Welling, PHP & My SQL, SAMS publications, 3rd edition.

COMPUTER AIDED SOFTWARE ENGINEERING TOOLS LAB

IT328 **Credits: 2**

Practical: 3 Periods /Week Sessional Marks: 50

End Exam: 3 Hours End Exam Marks: 50

Prerequisite:

Object oriented concepts, C++ programming, Fundamentals of Software Engineering

Course Objectives:

- Learn the basics of OO analysis and design skills
- Be exposed to the UML design diagrams
- Learn to map design to code
- Be familiar with the various testing technique

Course Outcomes:

After completion of this course, a student will be able to:	
1.	Design and implement projects using OO concepts
2.	Use the UML analysis and design diagrams
3.	Create code from design and contrast various testing techniques

Mapping of course outcomes with program outcomes:

		PO												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	2	2	3	3	3				3		3	2	2	2
	2	2	3	3	3	3				3		3	2	2	3
	3	2	3	3	3	3				3		3	2	2	3

LIST OF EXPERIMENTS

- To develop a mini-project by following the 9 exercises listed below:
- To develop a problem statement.
- Identify Use Cases and develop the Use Case model.
- Identify the conceptual classes and develop a domain model with UML Class diagram.
- Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
- Draw relevant state charts and activity diagrams.
- Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
- Develop and test the Technical services layer.
- Develop and test the Domain objects layer.
- Develop and test the User interface layer.

SUGGESTED DOMAINS FOR MINI-PROJECT:

- Web Content Management System
- Internet of Things
    - Speech Recognition
    - Enterprise Resource Planning (ERP) based system
    - Online Analytical Processing (OLAP)
    - Online Transaction Processing (OLTP)
    - Information Security
    - Image Processing
    - Automation Systems

*** Any other domain as per student's interest and instructor suggestion**

Suggested Software Tools:

- Rational Suite (or) Argo UML (or) equivalent, Eclipse IDE and Junit
- Visual Paradigm

Reference Books:

1. Grady Booch, the UML user guide.

SOFT SKILLS LAB

IT329 **Credits: 2**

Instruction: 3 Periods/week Sessional Marks: 100

Prerequisites:

Basic english language skills- LSRW. English language course and English Language Lab offered in the first year.

Course Objectives

- To inculcate effective communication skills with appropriate body language.
- To produce potent leaders, productive team players and effective individuals with proper professional ethics.
- To enable students to make successful oral presentations using relevant content.
- To train students for Group discussions and job Interviews which improves their employability skills.
- To make the students understand the importance of setting realistic goals and achieving them using time management techniques.

Course Outcomes

At the end of the course, students will be able to:	
1	Comprehend the core engineering subjects using effective verbal and nonverbal communication skills.
2	Present accurate and relevant information efficiently, using suitable material aids.
3	Work effectively as individuals as well as in teams and emerge as responsible leaders with appropriate professional ethics.
4	Participate in group discussions and interviews using analytical and problem solving abilities, which enhance their employability skills.
5	Set time bound goals and realize them through strategic plans for successful career.

Mapping of course outcomes with program outcomes:

	PO													PSO	
		1	2	3	4	5	6	7	8	9	1O	11	12	1	2
C O	1	1			2			3		3		1	1	1	
	2							3		3	3	1		2	
	3						3		2	3	2		2		2
	4		1					3		3	3		2	2	
	5			2						3			2		2

SYLLABUS

UNIT-I: Art of communication **9 Periods**

1. Definition of Communication
2. Types of Communication
3. Non-verbal Communication
4. Listening skills
5. Feed back

D.A. - Practice of proper hand shake, practice of different postures and gestures and activity on giving feedback

UNIT- II: Presentation Skills **6 Periods**

1. Purpose
2. Effective presentation strategies
3. Analysis of audience
4. Preparing an outline of the presentation,
5. Audio –visual aids
6. Body language**.**

D.A. -Group presentation by each team

UNIT- III: Group Discussions **9 Periods**

1. Introduction- as a part of selection process-guidelines for GD
2. Types of GD
3. Nature of topics of G.D
4. Roles to be played by participants in a GD
5. Evaluation process

D.A–Group discussions

UNIT – IV: Team Building and Leadership **6 Periods**

1. Importance of team work
2. Different stages of team formation
3. Good team vs. effective team
4. Team player and Team leader
5. Types of leadership
6. Decision making and negotiating skills

D.A-Decision making for a given situation

UNIT –V: Time- Management **3 Periods**

1. Importance of time-management
2. Time-Management models
3. Prioritization
4. The art of saying ‘No’
5. Identifying Time Wasters

D.A -Time- Bound activities devised by the facilitator

UNIT- VI: Goal-Setting **3 Periods**

1. Different type of Goals (Immediate and Short term)
2. 'SMART' Goals
3. Strategies to achieve goals

D.A - Prepare a chart of immediate, short term and long term goals

UNIT- VII: Job- Interviews **9 Periods**

1. Preparing Resumes and C.V's
2. Preparing for the interview
3. FAQ's (Integrity, Stress management, Close- Ask questions)

D.A –Mock interviews

Reference Books:

1. Sanjay Kumar and Pushpalata, ***Communication Skills***,Oxford University Press, 2011.
2. Allan Pease, **Body Language,** Sheldon Press,1997.
3. John A. Kline and BhavnaBhalla, ***Speaking Effectively; Achieving Excellence in Presentations***, Pearson publication, 2013.
4. Marc Mancini, ***Time Management***, Tata McGraw Hill publishing Comp.Ltd.,2003.
5. Peter Veruki, ***The 250 Job Interview Questions,***Adams Media Corporation Avon, Massachusetts,1999.

Quantitative & Verbal Aptitude - II
(Common for all Branches)

IT3210 **Credits: 2**

Instruction: 4 Periods/Week (2QA + 2VA) Sessional Marks: 100

Course Objectives:

Quantitative aptitude-II:

- To categorize, apply and use thought process to distinguish between concepts of reasoning
- To prepare and explain the fundamentals related to various possibilities and probabilities related to quantitative aptitude.``
- To critically evaluate numerous possibilities related to puzzles.

Verbal aptitude-II:

- To prepare the students on the various aspects of writing, organizing data, and applying their writing skills in their professional career.
- To demonstrate and recommend the techniques required when interacting in different situations.
- To apply the professional qualities/skills necessary for a productive career and to instill confidence through attitude building.
- To plan activities in order to expose students to the different abilities required for working in a team, encourage them to glean information on current affairs and promote factual reading.
- To illustrate and explain the intricacies/nuances involved in framing responses to the questions asked, reading between lines and reading beyond lines.

Course Outcomes:

Quantitative Aptitude-II

The student will be able to	
1	Use their logical thinking and analytical abilities to solve reasoning questions from company specific and other competitive tests.
2	Solve questions related to permutation & combinations and probabilities from company specific and other competitive tests.
3	Understand and solve puzzle related questions from specific and other competitive tests.

Verbal aptitude-II:

The student will be able to	
1	Write paragraphs on a particular topic, essays (issues and arguments), e mails, summaries of group discussions, make notes, statement of purpose (for admission into foreign universities), letters of recommendation (for professional and educational purposes)
2	Converse with ease during interactive sessions/seminars in their classrooms, compete in literary activities like elocution, debates etc., raise doubts in class, participate in JAM sessions/versant tests with confidence and convey oral information in a professional manner using reason
3	Prepare his/her resume, apply the business English concepts learnt in the course, and refine one's overall demeanor which would be very essential to face the corporate world
4	Prepare his/her resume, apply the business English concepts learnt in the course, and refine one's overall demeanor which would be very essential to face the corporate world
5	Respond to their interviewer/employer with a positive mind, customize answers to the questions asked during their technical/personal interviews, exhibit skills required for the different kinds of interviews (stress, technical, HR) that they would face during the course of their recruitment process

Mapping of Course Outcomes with Program Outcomes:

CO		**PO**												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
QA	1														
	2														
	3														
VA	4														
	5														
	6														
	7														
	8														

SYLLABUS

Section –A (Quantitative Aptitude –II)

UNIT I: **8 Periods**

Numerical Reasoning:

Problems related to Number series, Analogy of numbers, Classification of numbers, Letter series, Seating arrangements, Directions, blood relations and puzzle test.

UNIT II: **4 Periods**

Combinatorics:

Counting techniques, Permutations, Combinations and Probability

UNIT III: **4 Periods**

Data sufficiency Syllogisms

UNIT IV: **4 Periods**

Application of Base system:

Clocks (Base 24), Calendars (Base7), Cutting of Cubes and cuboids

UNIT V: **4 Periods**

Puzzle Solving & Time Management using various problems solving tools and techniques:

Selective puzzles from previous year placement papers

Selective puzzles from book Puzzles to puzzle you by shakunataladevi Selective puzzles from book more puzzles by shakunataladevi Selective puzzles from book puzzles by George summers

Books for practice

1. Quantitative aptitude by RS Agarwal, S Chand Publications
2. Verbal and non verbal Reasoning by RS Agarwal from S Chand publications
3. Puzzles to puzzle you by shakunataladevi orient paper back publication
4. More puzzles by shakunataladevi orient paper back publication
5. Puzzles by George summers orient paper back publication.

References:

1. Barron’s by Sharon Welner Green and Ira K Wolf (Galgotia Publications pvt. Ltd.)
2. Material from ‘IMS, Career Launcher and Time’ institutes for competitive exams.
3. Reasoning by BS Sijwali Arihant publications
4. Reasoning Arun Sharma McGrawhill publications

Websites:

1. www.m4maths.com
2. www.Indiabix.com
3. 800score
4. Official GRE site
5. Official GMAT site

Section –B (Verbal Aptitude –II)

UNIT I: **4 Periods**

General Essay writing, writing Issues and Arguments(with emphasis on creativity and analysis of a topic), paragraph writing, story writing, guidance in framing a 'Statement of purpose', 'Letters of Recommendation', business letter writing, email writing, email and business letter writing etiquette, letters of complaints/responses to complaints. Information transfer is taught with the help of tables, bar diagrams, and pie charts while framing /sending lengthy data where testing is done through Reading comprehension and Critical reasoning. Contextual meanings with regard to inflections of a word, frequently confused words, words often mis-used, words often mis-spelt, multiple meanings of the same word (differentiating between meanings with the help of the given context), foreign phrases. Enhanced difficulty level in spotting errors will be taken up with reference to competitive test based exercises.

UNIT II: **4 Periods**

Just a minute sessions, reading news clippings in the class, extempore speech, telephone etiquette, making requests/suggestions/complaints, elocutions, debates, describing incidents and developing positive non verbal communication. Analogies, YES-NO statements (sticking to a particular line of reasoning)

UNIT III: **4 Periods**

Corporate readiness, business idioms and expressions, reading newspapers/magazines, brushing up on general awareness, latest trends in their respective branches, resume preparation, understanding business /corporate language, managing emotions, problem solving, importance of team work, goal orientation, professional grooming, positive attitude, assertiveness and inter personal skills.Data sufficiency (answering questions within the ambit of the given text), Fact-Inference-Judgment (to identify statements as FIJ), Syllogisms (with emphasis on fallacies in reasoning), strong and weak arguments.

UNIT IV: **6 Periods**

Voice, direct & indirect speech, question tags, one word substitutes, and foreign phrases. An overview on group discussions, preparation for a group discussion, intricacies of a group discussion, topics for GDs (with special focus on controversial topics), structure of participation in a group discussion, roles played by the participants in a group discussion, constructive criticism, standard procedures followed whilst participating in a group discussion, frameworks that can be used for discussion, analysis of the discussion and exposure to case-based group discussions.

UNIT V: **6 Periods**

Different types of interviews (with emphasis on personal interview), preparation for an interview, areas of questioning, answering questions on general traits like strengths/ weaknesses/ hobbies/ extracurricular activities, choosing role models, importance of non verbal communication while participating in interviews, tips to reduce nervousness during personal interviews, handling stress, suggestions for responding to tough/unknown questions, preparation on self and personality development.

Note: The concepts learnt in Semester I will be tested in the Mid-term and Semester end exams during the II Semester as well.

Reading/ Listening material:

1. Newspapers like ‘The Hindu’, ‘Times of India’, ‘Economic Times’.
2. Magazines like Frontline, Outlook and Business India.
3. News channels NDTV, National News, CNN

References:

1. Books written by Stephen Covey and Dale Carnegie-Seven Habits ofHighly Effective People etc-Simon & Schuster, Running Press book publishers
2. Books written by Bertrand Russell-Oxford University Press

Suggested General Reading

1. **Who Moved My Cheese?** By Spencer Johnson-GP Putnam’s Sons
2. **The art of War**-Sun Tzu by Nabla, Barnes & Noble
3. **The Monk Who Sold Ferrari**-Robin Sharma by Harper Collins, Jaico Publishers
4. **The Hobbit** and other books by JRR Tolkein-Harper Collins

Suggested Authors

1. William Dalrymple
2. V.S.Naipaul
3. Kushwanth Singh
4. Ernest Hemingway
5. Charles Dickens
6. Leo Tolstoy
7. R.K. Narayan
8. Amitav Ghosh
9. Vikram Seth
10. Oscar Wilde