Board of Studies
Computer Engineering and IT.
Bharati Vidyapeeth University, Pune

STRUCTURE AND THE SYLLABI

B.TECH.

[COMPUTER ENGINEERING]

SEMESTER V AND VI

[2014 Course]

Bharati Vidyapeeth University, Pune
Faculty of Engineering & Technology
Programme: B. Tech (Computer Engineering) Sem – V (2014 Course)

Sr. No	Name of Course	Teaching Scheme			Examination Scheme							Credits		
		L	P	T	ESE	Continuous Assessment			Practical		Total	Theory	TW	Total
						Unit Test	Attendance	Assignment	TW PR	TW OR				
1	Formal Language and Automata Theory	3	--	1	60	20	10	10	--	--	100	4	-	4
2	Software Testing	3	2	--	60	20	10	10	--	50	150	3	1	4
3	Computer Network	3	2	--	60	20	10	10	50	--	150	3	1	4
4	Database Management System	3	2	--	60	20	10	10	50	--	150	3	1	4
5	Elective - I	3	--	--	60	20	10	10	--	--	100	3	--	3
6	Professional Skill Development-V	4	--	--	100	--	--	--	--	--	100	4	--	4
7	Programming Lab-III	--	4	--	--	--	--	--	50	--	50	--	2	2
	TOTAL	**19**	**10**	**1**	**400**	**100**	**50**	**50**	**150**	**50**	**800**	**20**	**05**	**25**

Elective - I: a) Multimedia and Mobile Applications b) Scripting Languages c) Software Project Management d) Computational Genomics

Optional Subject

Sr. No.	Name of Course	Teaching Scheme			Examination Scheme							Credits		
		L	P	T	ESE	Continuous Assessment			Practical		Total	Theory	TW	Total
						Unit Test	Attendance	Assignment	TW PR	TW OR				
	Engineering Mathematics IV	4	--	--	60	20	10	10	--	--	100	4	--	4

Bharati Vidyapeeth University, Pune
Faculty of Engineering & Technology
Programme: B. Tech (Computer Engineering) Sem –VI (2014 Course)

Sr. No.	Name of Course	Teaching Scheme			Examination Scheme							Credits		
		L	P	T	ESE	Continuous Assessment			Practical		Total	Theory	TW	Total
						Unit Test	Attendance	Assignment	TW PR	TW OR				
8	Operating System	3	2	--	60	20	10	10	--	50	150	3	1	4
9	Design and Analysis of Algorithm	3	2	--	60	20	10	10	--	50	150	3	1	4
10	Digital Signal Processing	3	2	--	60	20	10	10	--	50	150	3	1	4
11	Computer Organization and Architecture	3	--	1	60	20	10	10	--	--	100	4	--	4
12	Elective - II	3	--	--	60	20	10	10	--	--	100	3	--	3
13	Professional Skill Development-VI	4	--	--	100	--	--	-	--	--	100	4	--	4
14	Programming Lab-IV	--	4	--	--	--	--	--	50	--	50	--	2	2
	TOTAL	**19**	**10**	**1**	**400**	**100**	**50**	**50**	**150**	**50**	**800**	**20**	**05**	**25**

Elective - II: a) VLSI b) Natural Language Processing c) Human Computer Interaction d) Data Storage Systems

Total Credits Sem – III : 25
Total Credits Sem – IV : 25
Grant total : 50

FORMAL LANGAUGE AND AUTOMATA THEORY

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks	Theory:04 Credits
Tutorial: 1 Hours / Week	Continuous Assessment: 40 Marks	

Course Pre-requisites:

1. **Discrete Mathematics**
2. **Introductions to algorithms**

Course Objectives:

1. Learn types of grammars.
2. Understand various Computing models like Finite State Machine, Pushdown Automata, and
3. Turing Machine.
4. To Learn various applications of Formal Language Processing

Course Outcomes: After completion of course, students will able to:

1. Illustrate the importance of Automata Theory in designing computer languages.
2. Transform informal problems into formal ones.
3. Infer Grammars, languages modeling and compilers basics.
4. Investigate and prove the equivalence of languages described by pushdown automata
5. Design Language Acceptability by Turing Machine
6. Outline the applicability of the formal language and automata theory concepts.

UNIT I **Finite State Machine (FSM):** Introduction, difference between natural and formal languages, Basic machine - design of basic machines. Transition diagram, Transition graph, Acceptance of String, Acceptance of Language, examples, Finite Automata (FA), Deterministic Finite Automaton (DFA) and Non-Deterministic Finite Automaton (NDFA), Conversions of NFA with Є and without Є, Minimization of DFA, Equivalence of NFA and DFA, Limitations of FA. **(06 Hours)**

UNIT II **Regular expressions (RE) -** Introduction, FA and RE, RE to FA, FA to RE, algebraic laws, applications of REs, FA for regular grammar, Uses of Regular expression, Pumping Lemma. **(06 Hours)**

Moore and Mealy Machines: Introduction, Difference between Moore & Mealy, models, inter conversions, Equivalence of Mealy machine and Moore machine, Uses of both the machines.

UNIT III **Grammar-** Introduction, representation of grammar, Chomsky hierarchy, Context Free Grammar- Derivation, sentential form, inference, derivation, parse tree, ambiguity in grammar and language- ambiguous Grammar, removing ambiguity from grammar, Normal Forms- Chomsky normal form, Greibach normal form, Closure properties of CFL, Decision property of CFL, Reduced form grammar removal of unit productions, epsilon production, **(06 Hours)**

useless symbols. left linear and right linear grammars and inter conversions.

UNIT IV	**Push Down Automata (PDA):** Introduction, Pushdown Automata (PDA), Transition Diagrams, Functions and Tables, Deterministic Push- down Automata (DPDA) - definition, Nondeterministic Pushdown Automata (NPDA), Equivalence of context free grammars and PDA, properties of context free languages. Introduction to Post Machines (PMs)	**(06 Hours)**
UNIT V	**Turing Machine (TM):** Introduction, Transitions Diagrams, Functions and Tables, Design of TM as generator, decider and acceptor, comparison of Turing machine (TM) with FSM, PDM, and PM. combination TM, iterative TM, recursive TM, universal TM, Language Acceptability by TM, Recursive sets, partial recursive functions, recursively enumerable sets, Church's Turing hypothesis, multi stack Turing machine, TM limitations.	**(06 Hours)**
UNIT VI	**Applications** –Application of RE, FA, PDA, CFG, TM. Syntax analysis language definition. Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. Lexical analyzer, Text editor, and searching using RE. Introduction to Natural Language Processing.	**(06 Hours)**

Assignment:

Note:

For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.

The assignments are to be submitted as a hard copy.

Text books

1. E.V. Krishnamurthy, “Theory of Computer Science”, EWP Publication
2. Vivek Kulkarni “Theory Computation” Oxford higher education.

References:

1. Hopcroft Ullman, “Introduction to Automata Theory, Languages & Computations, Narosa
2. Daniel A. Cohen, “Introduction to Computer Theory”, Wiley Publication
3. John C. Martin, “Introduction to Language & Theory of Computation”, McGraw Hill
4. Mishra K L P and Chandrasekaran N, “Theory of Computer Science - Automata, Languages and Computation”, Third Edition, Prentice Hall of India, 2004

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

SOFTWARE TESTING

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Exam: 60 Marks	Theory: 03 Credits
Practical: 02 Hours / Week	Continuous Assessment: 40 Marks	TW/OR: 01 Credits
	Term Work/Oral: 50 Marks	

Course Pre-requisites:

The Students should have

1. Basic knowledge of Software Engineering.

Course Objectives:

1. Create awareness among the students about the basic fundamentals of Software Testing Process.

2. To make students aware of various techniques and tools used for testing different kinds of software.

Course Outcomes: After completion of course, students will able to:

1. Define Software Testing level, techniques, test deliverables/artifacts, V-Model.

2. Comprehend various Testing Levels and Techniques.

3. Compare different testing approaches for various kinds of applications.

4. Analyze business and software Risks involved in managing Software Testing Projects. .

5. Outline substantial knowledge of Agile Testing Process.

6. Recite Agile Testing Techniques and Tools.

UNIT I **Introduction:** **(06 Hours)**

Testing as an Engineering Activity – Basic Definitions – Bug, Defect, Verification validation, Debugging ,Software Testing Principles – The Tester's Role in software Development Organization , Iterative development – Risk Driven and Client Driven, Test driven Development, Evolutionary and adaptive development , Evolutionary requirements analysis – Early "Top Ten" high-level requirements and skillful analysis, Testing Fundamentals- Quality Assurance, Quality Control, V-Model of software testing, Testing techniques and Levels of Testing, Static Vs Dynamic testing, Test deliverables.

UNIT II **Testing Techniques and Test Case Fundamentals.** **(06 Hours)**

Testing Levels- Unit Testing, Integration Testing, Functional and System Testing, Types of Testing- Stress Testing, Performance Testing, Usability Testing, Non-Functional Testing, Acceptance Testing, Regression Testing, Beta Testing, Testing techniques- Black Box Vs. White Box Testing, Black Box Techniques, White Box Techniques, and Verification Techniques: Inspection, Walk-Through, Peer Reviews.

Test Case Fundamentals- Definition, Test Case Template, Test Case Parameters, Kinds of Test cases, Characteristics of Good Test Case, Writing Functional and Boundary Test Cases, Manual Test Cases Vs Automated Test Scripts.

UNIT III **Testing approaches for different Types of Applications** **(06 Hours)**

Testing COTS (Commercial-Off-the-shelf Software, Web-based applications/e-commerce applications, Object-Oriented Systems, Wireless applications, Testing for security, Website Testing, Foreign Language Testing.

Test Automation Tools: Web browser Automation through Selenium, Test

Management Tool-Test Director, Defect Tracking Tool – Bugzilla, GUI Testing Tool- WinRunner, and Configuration Management Tools.

UNIT IV **Managing Software Testing Projects** **(06 Hours)**

Test Planning- Test strategy, Creating a test Plan, Test Plan document, Test Pass/Fail criteria, Risk analysis, Test Scoping and Effort Estimation, Test Scheduling and budgeting. Introduction to Testing Maturity Model (TMM).

Test Metrics: Types, Identifying Test Metrics, Methods for gathering Test metrics, Analyzing and Applying Test Metrics, GQM Paradigm. Defect Management: Defect Tracking, Defect Reporting, Defect Metrics.

UNIT V **Agile Methodology** **(06 Hours)**

Introduction to Agile Model, Agile Vs Waterfall Method, Methodologies of Agile Testing - Scrum- Practices, Process flow of Scrum, eXtreme Programming- Phases, Crystal Methodologies, DSDM (Dynamic Software Development Method), Agile Testing Lifecycle. Test-Driven Development, Acceptance Test-Driven Development, and Behavior-Driven Development, Role of the Agile Tester, Assessing Quality Risks in Agile Projects, Techniques in Agile Projects- Acceptance Criteria, Adequate Coverage, Applying Acceptance Test-Driven Development, Exploratory Testing and Agile Testing, Tools in Agile Projects.

UNIT VI **Agile Testing Techniques and Quality Tools**: **(06 Hours)**

Agile testing – Nine principles and six concrete practices for testing on agile teams. Six Sigma - Kaoru Ishikawa's Basic Seven QC (Quality Control) Tools- Cause and Effect diagram, Check Sheet, Control charts, Histogram, Pareto chart, Scatter diagram, Flowchart, PDCA (Plan-Do-Check-Act) cycle, Software Testing- ISO Standards, IEEE Standards- IEEE 829, IEEE 1008, IEEE 1012 etc.

Term Work:

1. Introduction to Software Testing Lifecycle Model and the SPRAE Software Testing Framework.
2. Design functional and boundary test cases manually, execute tests, and evaluate test results.
3. Write manual test cases for a C program which demonstrates the working of the following
   A. Constructs: I) do...while II) switch
   B. A program written in C language for Matrix Multiplication fails." Introspect the causes for its failure and write down the possible reasons for its failure".
4. Write the test cases for any known application (e.g. Banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Consider any system (e.g. ATM system) and study its system specifications and report the various bugs.
7. Running test scripts using automated testing tool (e.g. Win runner)
8. Testing of a web application using the web testing tool (e.g. Selenium)
9. Tracking and reporting bugs using bug tracking tool (e.g. Bugzilla, bugbit)
10. Create a database of manual and automated tests using any open source test management tool.
11. Implement software testing on an Agile Project.

Assignments:

1. The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.
2. In case of assignments for internal 10 marks students will be assigned two assignments containing

problems of different types or any programming assignment and guided for the solutions of the problem.

3. The assignments are to be submitted as hard copy.

Text Books:

1. William.E. Perry, "Effective Methods for software Testing", Wiley 3rd Edition.
2. Ron Patton, "Software Testing", Techmedia.
3. Elfriede Dustin," Effective Software Testing", Addison-Wesley, 1st Edition, 2003.
4. Lisa Crispin, Janet Gregory," Agile Testing", Addison-Wesley, 11th Edition, 2015.

Reference Books:

1. Marnie Hutcheson," Software Testing Fundamentals: Methods and Metrics", Wiley.
2. Paul C. Jorgensen, "Software Testing: A Craftsman's Approach", Auerbach Publications, 2008.
3. Craig Larman, "Agile and Iterative Development – A Manager's Guide", Pearson Education – 2004.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

COMPUTER NETWORK

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks	Theory:03 Credits
Practical: 2 Hours / Week	Continuous Assessment: 40 Marks	TW/PR:01 Credit
	Term work/Practical: 50 Marks	

Course Pre-requisites:

1. Students should have basic knowledge of Computers and Internet.

2. C/C++/Java programming, algorithms & probability.

Course Objectives:

1. Developing the students with the knowledge of advanced computer networking.

2. Developing students with practical knowledge of latest networking technologies.

Course Outcomes: After completion of course, students will able to:

1. Recite the basics of Computer network.

2. Relate detailed structure of data link layer with other layers.

3. Enumerate the concept of Medium Access Control layer.

4. Recite the details of Network layer.

5. Discuss the details of Transport layer.

6. Infer the functionality of Application layer.

UNIT I **INTRODUCTION:** **(06 Hours)**

Internet working, Use of the computer network, Physical layer, Networking hardware devices, networking software, types of networks, OSI model working, TCP/IP reference model, Wireless networks, Buffering, Switching, packets, frames, introduction to latest internet technologies.

UNIT II **Data Link Layer:** **(06 Hours)**

Data layer working, Framing, Error Control, Flow Control, error detection and correction, data link protocols, IEEE standards.

UNIT III **MEDIUM ACCESS CONTROL SUBLAYER:** **(06 Hours)**

MAC layer working, Ethernet, Static & Dynamic Channel Allocation, multiple access protocols, wireless technologies, RFID.

UNIT - IV **NETWORK LAYER:** **(06 Hours)**

Network layer working, Network layer design issues, routing algorithms, congestion control algorithms, quality of service, IP Addresses, Subnets, Configuring network settings, Firewalls, IP V6, Mobile IP V6.

UNIT V **TRANSPORT LAYER:** **(06 Hours)**

Transport layer working, services of transport layer and elements of transport protocols, congestion control in transport layer, Transport protocols-TCP &UDP, Performance issues, Mobile TCP.

UNIT VI **APPLICATION LAYER:** **(06 Hours)**

Application layer working, DNS, Email, WWW, Audio & video streaming, Content delivery, Caching in Web Browser remote login, Wireless web, browsers, NFS, SNMP, Telnet, FTP, HTTP, WiMAX, WSN.

Term Work:

- Introduction and configuration of networking devices
  Practical Study of PC, router, Switches, hubs, servers, repeaters, Wi-Fi modem and its configurations
- Introduction to "CISCO's Packet tracer".
  Working and study of CISCO's Packet tracer
- Implementation of Packet switching using "CISCO's Packet tracer" software.
- Client-Server setup. Connection establishment between client and server. Study of FTP and HTTP through this setup. Study of packet delivery. Simulation of packet delivery.
- Implementation of static routing using "Packet tracer" software.
- Distance vector routing algorithm. Packet switching using static routing algorithm
- Implementation of dynamic routing using "Packet tracer" software.
- Dijkstra's shortest path algorithm. Packet switching using dynamic routing algorithm.
- WAP in C to implement routing algorithm using Bellman Fords distance vector algorithm.
- Study of network throughput and efficiency using "Wireshark software"
- Packet delivery from one system to other. Observe its throughput by using Wireshark software.
- Data encryption using "GnuPG" Software.
- Encryption of the data using GnuPG software. Encrypt data and deliver it to other system and
- Observing the change in file size using Wireshark
- Experiment related to NS2/NS3 tool.
- Case study of "Networking commands in Linux based operating system"

Assignment:

NOTE:

The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.

For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.

The assignments are to be submitted as a hard copy.

Text Books /Reference books

1. James f. Kurose, Keith w. Ross. Computer networking: a top-down approach. 6th ed.: Pearson.
2. Andrew s. Tanenbaum, David j. Wetherall.Computer networks.5th ed.: Pearson.
3. Forouzan. Data comm.& netw.5e. 5th ed.: McGraw-Hill.
4. William Stallings. Data and computer communications. 8th ed.: Pearson.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

DATABASE MANAGEMENT SYSTEM

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Examination: 60 Marks	Theory:03 Credits
Practical: 02 Hours / Week	Continuous Assessment: 40 Marks	TW/PR:01 Credit
	TW/Practical: 50 Marks	

Course Pre-requisites:

1. Discrete mathematics and Data structures.
2. Programming languages.

Course Objectives:

1. To provide a strong formal foundation in database concepts, technology and practice.
2. To design and develop a database schema.
3. To understand Structured query language (SQL), indexing and query processing.

Course Outcomes: After completion of course, students will able to:

1. Differentiate significance of Database Management System over the file processing system.
2. Illustrate the fundamentals of data models and to conceptualize and depict a database system using data models.
3. Analyze and practice Relational Data Model.
4. Apply SQL queries for database definition and database manipulation.
5. Illustrate transaction management concepts like serializability, concurrency control and recovery system.
6. Investigate the knowledge about emerging trends in the area of database for unstructured data and applications for it

UNIT I **Overview of Database Management System** **(06 Hours)**

Overview of DBMS: Data and Information, Database, characteristics of databases, Data Governance and Importance of database, Database Management System (DBMS), structure of DBMS, Objectives of DBMS: Data Availability, Data Integrity, Data Security, and Data Independence. Three level DBMS architecture and Data Abstraction. Classification of DBMS, File based System; Drawbacks of File based System, Advantages of DBMS.

Database Architecture: Two Tier, Three Tier, Multi-Tier. The DBMS Life Cycle, Information Life Cycle, Roles in Database Environment, Database users and Administrator, Introduction to Data models, Need for abstraction, Situation where DBMS is not necessary, DBMS vendors and their products.

UNIT II **Data modeling and Design** **(06 Hours)**

Data modeling: Benefits of Data Modeling, Types of Data Modeling, Phases of Data Modeling, Building Blocks of Entity Relationship(E-R) model, Mapping Constraints, Keys, Weak and strong Entity Sets, Extended-R features, Reduction to relational schemas, a case study on building an ER Model.

Database design: Objectives of Database design, Database Design Tool, Features of Design Tool, merits and De-merits of database design tool, Different anomalies in designing a Database, Functional Dependency, Inference Rules, Closure of set

Functional Dependency, Purpose of Normalization, Steps in Normalization: 1Normal Form(NF), 2NF, 3 NF Boyce-Codd normal form, denormalization: Types of De-normalization, Table De-Normalization Algorithm.

UNIT III **Relational Model and Algebra** **(06 Hours)**

Relational Data Model: CODD's Rule, Structure Part, Integrity Part, Manipulative, Table & Relation. Concept of Key: Super Key, Candidate Key, Foreign Key. Relational Integrity: Entity Integrity, NULL Integrity, Domain Integrity constraint, Referential Integrity, Data structure, Mapping ER model to Relational model.

Relational Algebra: Unary and Binary Operations, Rename Operation, Union Operation, Intersection, Difference, Division, Cartesian Product, Join Operations. Advantage and Limitation of Relational Algebra.

UNIT IV **SQL and Introduction to PL/SQL** **(06 Hours)**

SQL: SQL fundamentals, Data Definition Language(DDL), Data Manipulation Language(DML) and Data Control Language(DCL), Basic structure of SQL queries, set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, Sub queries, joins, Nested Sub queries, Complex queries, views: Creating, Dropping, Updation using Views.

PL/SQL: Introduction to PL/SQL block, Concept of Cursors, Stored Procedures, Triggers, Java Database Connectivity (JDBC), Open Database Connectivity (ODBC).

UNIT V **Transaction Processing and Query Optimization** **(06 Hours)**

Transaction processing: Concept of Transaction and its Management, concurrency Anomalies, Schedules, schedules and recoverability, Serializability, Hierarchy of serializable Schedules, Concurrency control and enforcing serializability, Deadlocks, Lock Granularity, Lock based Concurrency control, Multiple Granularity and Intension Locking, Time stamping Control, Optimistic Control, Evaluation of Concurrency Control Mechanism.

Query Processing and Optimization: Issues in Query Processing, Steps in Query Processing, Query decomposition, Query Optimization: Heuristic query optimization, Transformation rules, Heuristics optimization algorithm. Cost estimation in query optimization: Cost of components of query execution, cost for select and join operation, Query execution Plan.

UNIT VI **Data Warehousing and Data mining** **(06 Hours)**

Data Warehousing: Introduction, Evolution of Data Warehouse, Characteristics, and Benefits, Limitation of Data Warehousing, Main Components of Data Warehouse, Conceptual Models, Data Mart, and Online Analytical Processing (OLAP), Teradata Relational Database Management System(RDBMS), Teradata Technology.

Data Mining: Data Mining Concepts, Knowledge Discovery, Goals of Data Mining, Data Mining techniques, Machine learning using WEKA tool.

Emerging Database Technologies: Introduction to unstructured data, NOSQL, spatial and geographic databases, multimedia databases, Massive Datasets and Hadoop.

Term Work:

The sample practical assignments are given below. This can be used as a guideline and course coordinator can recommend the list of practical assignments.

1. Write a simple PL/SQL program to check whether the given number is palindrome or not and insert into a table reverse if the given number is palindrome?

2. To import various schemas into database system by running the scripts. Schemas are human resource, Order entry, Product Media, Queued shipping, Sales history.
3. To apply the select statements for the given queries.

   a. Display employees those who make more than $900 or if their names fall between QUENTIN and ZYRYAB alphabetically.

   b. Display employees those who make more than $900 and if their names fall between QUENTIN and ZYRYAB alphabetically.

   c. Display the name and job title of all employees who do not have a manager.

   d. Display the name, salary and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

4. To apply the single column functions:
   a. WAQ that displays the employee's names with the first letter capitalized and all other letters lowercase and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employee's name.
   b. For each employee, display the employee's name and calculate the number of months between today and the date employee was hired. Label the column months worked. Order your results by the number of months employed. Round the number of months up to the closest whole number.
   c. WAQ that produces following for each employee.
      i. <Employee name> earns <salary> monthly but wants < 3 times salary>. Label the column dream salaries

5. To discuss normalization and build normalized schema of Hospital Management system.
6. To demonstrate queries on Joins.

   a. Display the employee name and employee number along with their manager's name and manager number.

   b. To display all employees including king, who as no manager. Order the results by employee number.

   c. Display employee name, department name and all the employee who work in the same department as a given employee.

   d. Create a query to display the name and hire date of any employee hired after employees Davies.

7. To demonstrate queries on aggregate functions
   a. Determine the number of managers without listing them.

   b. Display the Manager number and the salary of the lowest paid employee for that manager.

   c. Display each department name, location, number of employee and the average salary for all employee in that department.

   d. Create a query that displays total number of employees and of that total, the number of employees hired in 1995, 1996, 1997 and 1998.

8.Use WEKA tool to derive analytical model for the given dataset.
9. Apply the association rules on the dataset and derive the inferences from the results given by WEKA tool.
10. Case study on NOSQL database: MongoDB.

Assignment:
NOTE:
The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.
For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.
The assignments are to be submitted as a hard copy.

Text books:
1) Silberschatz A., Korth H., Sudarshan S, "Database System Concepts", 6th Edition, McGraw Hill Publishers.
2)Ramakrishna R., Gehrke J., "Database Management Systems", 3rd Edition, McGraw- Hill.

Reference books:
1)Elmasri R., Navathe S., "Fundamentals of Database Systems", 5th Edition, Pearson Education.
2) Ryan K. Stephens, Ronald R. Plew,"SQL", 4th Edition, Pearson Education.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

ELECTIVE – I: MULTIMEDIA AND MOBILE APPLICATIONS

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Exam: 60 Marks Continuous Assessment: 40 Marks	Theory: 03 Credits

The Students should have

1. Basic knowledge of JAVA Programming.
2. Basic knowledge of Data Communication.

Course Objectives:

1. Create awareness among the students about the basic fundamentals in the design and development of multimedia systems and applications.
2. To make students aware of various techniques used for the development of distributed applications for mobile devices as well as introduce them to the design principles for applications for small devices

Course Outcomes: After completion of course, students will able to:

1. Recite basic building blocks of Multimedia systems and applications.
2. Differentiate between various Compression and Decompression Techniques.
3. Comprehend various Multimedia I/O, Storage and Retrieval technologies.
4. Analyze various issues pertaining to Multimedia Application Design.
5. Report on Multimedia Networks.
6. Discuss on Mobile Application Development.

UNIT I **Introduction to Multimedia:** **(06 Hours)**

Multimedia- Building Blocks, Overview of Multimedia Applications, Multimedia Systems- Components, Characteristics, key issues and Challenges, Evolving Technologies for Multimedia Systems, Multimedia Streaming Protocols (MSP).
Multimedia Data: Text and static data, Graphics, Images, audio, video. Multimedia Data Compression- Lossy and Lossless compression techniques.

UNIT II **Compression and Decompression Techniques** **(06 Hours)**

Types of Compression, Binary Image Compression Schemes, Color, grayscale, still-video image compression, Discrete Cosine Transform, Video image compression, MPEG Coding methodology, Audio Compression, Data and File format standards- PDF, RTF, TIFF, RIFF, MIDI, JPEG, AVI, JPEG, MPEG.

UNIT III **Multimedia I/O, Audio and Video** **(06 Hours)**

Key Technology Issues, Pen Input, Video and Image Display Systems, Print Output Technologies, Image Scanners, Digital Voice and Audio, Video Images and Animation, Full Motion Video, Magnetic Media Technology, WORM optical drives, Cloud-Based Multimedia Storage systems.
Multimedia Databases: Design and Architecture of a Multimedia Database, Organizing Multimedia Databased on The Principle of Uniformity, Media Abstractions, Query Languages for Retrieving Multimedia Data.

UNIT IV **Multimedia Application Design** **(06 Hours)**

Types of Multimedia systems - Virtual Reality Design - Components of Multimedia

system - Distributed Application Design Issues - Multimedia Authoring and User Interface - Hypermedia Messaging- Distributed Multimedia Systems, Multimedia Authoring Tools.

UNIT V	**Multimedia Networks** Basics of Multimedia Networks, Multimedia Network Communications. Applications: Quality of Multimedia Data Transmission, Multimedia over IP, Multimedia over ATM Networks, Transport ofMPEG-4, Media-on-Demand(MOD). Video Broadcasting Standards- HDTV, 4K TV and Ultra HD, Multimedia Content Management (MCM).	**(06 Hours)**
UNIT VI	Introduction: Understanding Mobile Platforms, Android as Competition to itself, Building an App in Android, Debugging Android Apps. Building block of Mobile apps: App user Interface Designing, Layout, User Interface elements, Activity states and lifecycle, Mobile Databases such as SQLite and enterprise data access, Windows Mobile OS (Operating System).	**(06 Hours)**

Assignments:

NOTE:

The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.
In case of assignments for internal 10 marks students will be assigned two assignments containing problems of different types or any programming assignment and guided for the solutions of the problem.
The assignments are to be submitted as hard copy.

Text Books:

1. Ralph Steinmetz, Klara Nahrstedt –Multimedia computing, Communication & Application– Pearson Education, 6th Edition.
2. Ze-Nian Le, Mark.S. Drew, Jiang Chuan Liu, Fundamentals of Multimedia, Springer, 2nd edition.
3. JakobIversen, Michael Eierman, Learning Mobile App Development- A Hands-On Guide to Building Apps with iOS and Android, Addison-wesley,1st edition, 2013.

Reference Books:

1. Tay Vaughan, "Multimedia making It work", Tata Mc Graw Hill 5th Edition 2001.
2. Fred Halsall, "Multimedia Communications", Addison Wesley, 1st Edition, 2000.
3. Andleigh P K and Thakrar K," Multimedia Systems", Addison Wesley , 1999.
4. Jeff Mc Wherter, Scott Gowell, "Professional Mobile Application Development", Wiley Publications.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

ELECTIVE – I: SCRIPTING LANGUAGES

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks Continuous Assessment: 40 Marks	Theory:03 Credits

Course Pre-requisites:

1. The course prerequisites are some general familiarity with programming language constructs such as loops, functions, and conditionals. And revision of regular expressions.

Course Objectives:

1. The study of the principles of scripting languages
2. Understand the difference between client side and server side scripting.
3. Understand the basic techniques used to create scripts for automating system administration tasks.

Course Outcomes: After completion of course, students will able to:

1. To Exemplify the fundamentals of scripting languages.
2. To design and implement JavaScript programs using web browser.
3. To analyze the DOM model and JSON.
4. To implement JQuery.
5. To demonstrate basics of PHP and implement the program using web server.
6. To exhibit the fundamentals of python script and implement it using python interpreter.

UNIT I **Introduction to Scripting Languages**: Motivation for and applications of Scripting languages; Difference between Scripting languages and non-Scripting languages; Merits of scripting languages; Types of Scripting languages, Client Side Scripting, Scripting for content structuring, Form design, Client Side Validation, dynamic page generation, adding interactivity, Styles, using HTML, DHTML, XHTML, CSS, Java Script. Server Side Scripting, Types of servers - Configuring and Using Web servers, Important features and Examples of other Scripting Languages. **(06 Hours)**

UNIT II **Java Script fundamentals& Working with Java Scripts:** **(06 Hours)**
Introduction to Java Script, History, Variables, Data Conversions, Interaction with the User, Operators, Math and Date Objects, Flow control, String Objects, String Methods, Number Objects, Formatting Numbers, Common Array Methods, Associative Arrays.
Implementing Functions: Declaration & Invocation, Passing Arguments, Returning Values, Anonymous Functions and Event Handling.

UNIT - III **JavaScript Objects and JSON:** **(06 Hours)**
Browser Object Model, Document Object Model (DOM), JSON, jQuery Selectors and Filters, DOM Manipulation, jQuery Events, jQuery Event Model, Common jQuery Events**,** Delegated and Direct Events. jQuery Plugins, Ajax Requests Using jQuery.

UNIT - IV **XML:** Creating Markup with XML, Document Type Definition, Schemas Document Object Model, Simple API for XML, Extensible Stylesheet languages, Formatting Objects, Xpath, XLink and Pointer, Introduction to SOAP, Case Studies, Custom markup languages. Parsers: DOM and SAX - Evolution of AJAX. **(06 Hours)**

UNIT - V	**Python Script:** Introduction, Conditional Statements, Looping, Control Statements, String Manipulation, Lists, Tuple, Dictionaries, Functions, Modules, Input-Output, Exception Handling.	**(06 Hours)**
UNIT - VI	**Python Script with OOP**: OOPs concepts, Classes in Python, Principles of Object Orientation, Instance Methods, File Organization, Special Methods, Class Variables, Inheritance, Polymorphism, Type Identification, Regular expressions, CGI, Database, Networking, Multithreading, GUI Programming.	**(06 Hours)**

Assignment:

Note:

The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.
For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.
The assignments are to be submitted as a hard copy.

Text books:

1)Python: The Complete Reference by Martin C. Brown

2) JavaScript: The Complete Reference by Thomas Powell, Fritz Schneider.

Reference Books:

1) JavaScript in 24 Hours, 4th ed. Michael Moncur. 2007. Sams Publishing.

2) Python Tutorial by Guido van Rossum, and Fred Drake, Jr., editor, Release 2.6.4.

3) Programming Python, by Mark Lutz.O'REILLY

4)Xml: The Complete Reference by Willimson, Tata McGraw-Hill Education

5) Beginning Python: From Novice to Professional (2nd Edition) Author: Magnus Lie Hetland.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

ELECTIVE – I: SOFTWARE PROJECT MANAGEMENT

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Examination: 60 Marks Continuous Assessment: 40 Marks	Theory:03 Credits

Course Pre-requisites:

1. Understanding of Software Development Life cycle(SDLC)
2. Knowledge of management skills and strategies.

Course Objectives:

1. Articulate similarities and differences between IT projects and other types of projects
2. The ability to come up with a project schedule and assign resources
3. Identify project risks, monitor and track project deadlines.
4. The capability to work in a team environment and be aware of different modes of communications

Course Outcomes: After completion of course, students will able to:

1. Practice the process of project management and its application in delivering successful IT projects.
2. Distinguish between the different types of project and follow the stages needed to negotiate an appropriate scope.
3. Plan and Complete the tasks in time effectively and efficiently.
4. Evaluate a project to develop the scope of work, provide accurate cost estimates and to plan the various activities.
5. Identify the resources required for a project and to produce a work plan and resource schedule.
6. Implement the project plans through managing people communications

UNIT I **An overview of IT Project Management** **(06 Hours)**

Introduction, the state of IT project management, context of project management, need of project Management, project goals, project life cycle, information technology project methodology (ITPM), project feasibility, project selection and approval, project contracting, PMBOK.

UNIT II **Project Integration & scope Management** **(06 Hours)**

Project management process, project Integration management, the project charter, project Management planning framework, the contents of a project plan, the planning process, project scope definition, Management project scope verification, change control, Work Breakdown Structure (WBS), linear responsibility chart.

UNIT III **Project Time Management:** **(06 Hours)**

Developing the project schedule, Management Scheduling Charts, logic diagrams and network (AOA, AON), critical path, calendar scheduling and time based network, PDM network, PERT, CPM, Resource loading, resource leveling, allocating scarce resources to projects.

UNIT IV **Project Quality and Cost Management:** **(06 Hours)**

Quality tools and philosophies, Quality management system, IT Project quality plan. Cost estimating, Cost escalation, Management system development cycle, Cost estimating process, Elements of budgets and estimates, Project cost accounting and MIS, Budgeting using cost accounts, Cost schedules and forecasts.

UNIT V **Project Human Resource Management:** **(06 Hours)**

Organization and project planning, Resource project team, multidisciplinary teams, project Management environment, project leadership, ethics in projects, multicultural projects, Role of project manager, IT governance.

UNIT VI **Project Human Communication Management:** **(06 Hours)**

Monitoring and controlling the project, Communication project plan, Project metric, Project Management control, designing the control system, the plan-monitor control cycle, data collection and reporting, reporting performance and progress.

Assignment:

Note: For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.

The assignments are to be submitted as a hard copy

Text books

1., Hughes, cotterel, RajibMalll Tata McGraw Hill Software Project Management.

2. Edwin Bennatan Software Project management.

References

1. S.A. Kelkar Software Project Management

2. Whitten, Bentley and Dittman System Analysis and Design Methods

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

ELECTIVE – I: COMPUTATIONAL GENOMICS

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks Continuous Assessment: 40 Marks Total: 100 Marks	Theory:03 Credits

Course Pre-requisites:

In order to understand the content, gain knowledge and successfully complete this course, students should have a basic understanding of the computer algorithms, Programming skills and basic mathematics.

Course Objectives:

1. The purpose is to enable students to analyze and Interpret data generated by bioinformatics/genomics technology.
2. Enable student to use statistical concepts to design experiments and analyze high dimensional data.

Course Outcomes: After completion of course, students will able to:

1. Apply knowledge of data structures, algorithms and analysis of algorithms to problems in Computational Genomics
2. Recite and Express Knowledge needed to read and interpret cutting-edge results in computational genomics.
3. Express the strengths and limits of current genomics data analysis methods
4. Prepare to lead new research projects in computational genomics
5. Formulate and/or model a biological problem/system as a computer science problem
6. Investigate the applicability of algorithms & techniques in other domains such as text mining, speech recognition, pattern matching and string searching

UNIT I **Introduction to Genome Computational** **(06 Hours)**

Introduction to Challenges in Computational biology: gene finding, Sequence alignment, Database lookup, genome assembly, Regulatory Motif Discovery etc.,
Introduction to Probability: Probability Distribution, Multiple Random Variables, Random Variables assuming infinite values.
Introduction to Markov & Chebycheff inequalities, Hoeffding's inequality, Monte carlo simulation, Cramer's theorem.

UNIT II **Algorithms for computational genomics** **(06 Hours)**

Enumeration Approaches: Exhaustive search, Pruning, greedy algorithms, iterative refinement.
Content based indexing: hashing, database lookup, pre-processing.
Iterative Method: Combining sub problems, memorization, dynamic programming.
Statistical Method: Hypothesis testing, Maximum likelihood, Bayes Law, HMM's.
Machining learning technique: Supervised and unsupervised learning, classification.

UNIT III **Hidden Markov Processes** **(06 Hours)**

Markov Processes: Markov property and state transition matrix, estimating the

state transition matrix, stationary Markov chains: recurrent and transient states, hitting probabilities and mean hitting time, Basic properties: Three different looking models, equivalence between three models. The Viterbi algorithm, Baum-Welch Algorithm.

UNIT IV **BLAST THEORY** **(06 Hours)**

Anatomy, Problem formulation, Moment generating function, Finding local Matches, Application of main results, Proofs of main results.
BLAST & Database: W-mer Indexing data, Search algorithm, karlin-altschul statistics, Filtering, Two hit Blast, substitution matrices, pigeonhole principle.

UNIT V **Data Mining** **(06 Hours)**

Functional Genomics: Splicing & Alternative Splicing, Microarray based Functional genomics.
Data Transformation: data smoothing by discretization, Normalization and standardization, min-max normalization, z-score standardization, use decimal scaling.
Feature Selection: Filter Approaches, Wrapper Approaches.
Clustering Technique: Distance based clustering and measures, K-means Algorithm, k-modes algorithm, Genetic Distance measure, hierarchical clustering, Graph based clustering.

UNIT VI **Classification in Genome and bioinformatics** **(06 Hours)**

Bias variance tradeoff in supervised learning, Linear and Nonlinear classifiers, Model complexity and training data set size, support vector machines, Bayesian Approaches, Bayesian Network, Decision tree: tree pruning.
Applications: Sequence alignment using dynamic programming, Scoring matrices for protein sequences.

Assignments:

Note

For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.
The assignments are to be submitted as a hard copy.

Reference Books

[1] Data Mining for Bioinformatics, Sumeet Dua, Pradeep Chowriappa, CRC Press, 06-Nov-2012
[2] Data Mining in Bioinformatics, Jason T. L. Wang, Mohammed J. Zaki, HannuToivonen, Dennis Shasha
[3] Springer Science & Business Media, 30-Mar-2006
[4] Hidden Markov Processes: Theory and Applications to Biology, M. Vidyanagar Princeton University Press, 24-Aug-2014
[5] Theoretical and Computational Methods in Genome Research, Sandor Suhai, Springer Science & Business Media, 06-Dec-2012
[6] Computational Genome Analysis: An Introduction, Richard C. Deonier, Simon Tavaré, Michael Waterman, Springer Science & Business Media, 13-Aug-2007

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

ENGINEERING MATHEMATICS-IV (OPTIONAL SUBJECT)

TEACHING SCHEME:		**EXAMINATION SCHEME:**	**CREDITS ALLOTTED:**
Theory:	04 Hours / Week	End Semester Examination: 60 Marks	04 Credits
Practical:	-- Hours / Week	Continuous Assessment: 40 Marks	

Course Pre-requisites:

The Students should have knowledge of

1. Determinants

2. Matrices

3. Differentiation

4. Integration of functions

5. Differential equation

Course Objectives:

The course aims at making the students familiar about the most basic numerical methods and concepts like error estimation helpful in various fields of engineering and can be used to simulate the results of various numerical methods.

Course Outcomes:

The student should be able to

1. Derive appropriate numerical methods to solve algebraic and transcendental equations

2. Evaluate the accuracy of common numerical methods.

3. Develop appropriate numerical methods to solve a difference equation.

4. Be familiar with numerical interpolation and approximation of functions, numerical integration and differentiation.

5. Be familiar with numerical solution of ordinary differential equations.

6. To compute Numerical Solution of Partial Differential Equations.

UNIT - I **Numerical solutions of algebraic and transcendental equations:** **(08 Hours)**

Bisection method, Regula-Falsi method, Newton-Raphson method, Direct iterative method.

UNIT - II **Solution of system of linear algebraic equation** **(08 Hours)**

Matrix inversion method, Gauss- elimination Method, Jordan's method, Crout's method. Gauss-Seidel and Gauss Jacobi's iterative method.

UNIT - III **Difference equation and Solution of difference equations** **(08 Hours)**

Definition of difference equations, formation of difference equation. Solution of Homogeneous and non-homogeneous difference equation with constant and variable coefficients using Boole's operator method and generating functions. Simultaneous difference equation.

UNIT - IV **Interpolation and Numerical differentiation and integration** **(08 Hours)**

Finite difference operator, Interpolation formula with equal and unequal intervals. Divided differences and central differences. Curve fitting : Method of least squares. Straight line, Second degree, parabola, Exponential curve.

Differentiation using forward, backward and divided difference General quadrature formula, Trapezoidal rule, Simpson's 1/3rd rule, Simpson's 3/8th rule, Weddle's rule.

UNIT - V **Numerical solution of I order ordinary differential equation** **(08 Hours)**

Solution by Euler's,method Euler' Modified method Taylor's series. Runga-kutta method. Milne's Predictors and Correctors method.

UNIT - VI **Numerical Solution of Partial Differential Equations** **(08 Hours)**

Classification of second order partial differential equations, Solution of Laplace's, Poisson's, heat and wave equations by finite difference methods, Use of method of characteristics for solution of initial and boundary value problems.

Text Books:

1. Gupta P.P.& Malik G.S., *Calculus of Finite Differences and Numerical Analysis*, Krishna Prakashan Mandir,Meerut, 21/e, 2006.
2. B.S.Grewal, *Engineering Mathematics*, Khanna Publishers, 12/e, 2006.

Reference Books:

1. Francis J. Scheid, Schaum's *Outline of Numerical Analysis*, McGraw-Hill, New York, 1989.
2. S. S. Sastry, *Engineering Mathematics*, Vol I, II Prentice Hall Publication, 3/e, 2004.
3. C.Ray Wylie & Louis C. Barretle, Advanced Engineering Mathematics, Tata McGraw Hill Publishing Co Ltd., 6/e,2003.

Syllabus for Unit Test:

Unit Test -1	UNIT – I,II,III
Unit Test -2	UNIT – IV,V,VI

Programming Lab-III

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Practical: 04 Hours / Week	Term Work/Practical: 50 Marks	TW/PR: 02 Credits

Course Pre-requisites:

The Students should have

1. Basic knowledge of object oriented programming.

Course Objectives:

1. To develop ability to use the computational languages necessary for engineering practice.

Course Outcomes: After completion of course, students will able to:

1. Recite and Express fundamentals of Visual Basic .NET Programming.
2. Illustrate Object Oriented Programming in VB.NET.
3. Demonstrate Windows Forms Application.
4. Apply the concept of developing Console Application.
5. Illustrate Inheritance and Polymorphism.
6. Experiment the data access using ADO.net.

UNIT - I Getting Started with Visual Basic .NET: **(06 Hours)**

Introduction to Visual Basic .NET, MS. Net Project, MSIL, JIT, .NET frame
work class library.
Module and Variable: Module and Namespace, Variables, array and structure.
Control flow and Error Handlings: Execution flow control. commands, functions & constants. Error Handling, Debugging .NET applications.

UNIT - II Object Oriented Programming using VB.NET: **(06 Hours)**

Methods, properties, constructors, object lifetime, events, Inheritance, in VB.NET., Interfaces, Attributes.
Programming in .NET Framework: Array, Lists and collection file, directories and streams.

UNIT - III Windows Forms Application: **(06 Hours)**

Windows forms Applications, Controls, Data Access in visual basic .NET, ADO.NET, Database connectivity.

UNIT - IV C#.net Language Basics **(06 Hours)**

Datatypes, Common Type System, Reference Type and Value Type, Variables Declaration, Concept of Class and Object.

Implicit and Explicit Casting, casting between other datatypes, Boxing and Unboxing, Enum and Constant, Operators, Control Statements, working with Arrays, working with Methods - Pass by value, Pass by reference.
Developing Console Application
Introduction to Project and Solution in Studio, compiling a C# program, Compiling and Building Projects, Using Command Line Arguments, Importance of Exit code of an application.

UNIT - V Inheritance, Polymorphism and Interface in C#.NET **(06 Hours)**

Exception Handling
Exception, Rules for Handling Exception, Exception classes and its important properties, use of try and catch, throwing exceptions, Importance of finally block.
Introduction to Operator Overloading, DLL, DLL Vs Exe., Types of DLL, concept and use of Class Library, Namespace.
WinForms
Introduction to Windows Forms, Controls, Menus and Context Menus, MenuStrip, ToolbarStrip, Graphics and GDI, SDI and MDI Applications, Dialogbox (Modal and Modeless), Form Inheritance.

UNIT - VI Data Access using ADO.NET **(06 Hours)**

Introduction to SQL. Creating Database using VS.NET, Establishing Connection with Database. Executing simple Insert, Update and Delete Statements, Executing Select Statement and using SqlDataReader
Data Access using ADO.NET – Dataset, Advantages of Dataset, Concept of Data Adapter, Data Table, DataGridView, Data Row, Adding / Editing / Deleting rows in the Data Table, Working with Data View.
Multithreading: Creating and Managing Threads, Threads Priority, Thread States, Thread Synchronization & Inter-thread Communication., Using Monitor
Delegates & Events: Delegate Declaration, Sample Application.

Term Work:

Concern Staff should frame assignments on Each Unit.

Text Books:

Jesse Liberty Learning Visual Basic .NET, O'Reilly.

Andrew Troelsen Pro C# 2008 and the .NET 3.5 Platform, Apress

Reference Books:

Steven Holzner,VB.NET Programming Black Book, Dreamtec Publications.

Matt Telles, C#Programming Black Book, Dreamtec Publications.
Syllabus for Unit Test: NA

OPERATING SYSTEM

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks	Theory: 03 Credits
Practical: 2 Hours / Week	Continuous Assessment: 40 Marks	TW/OR:01 Credit
	TW/Oral: 50 Marks	

Course Pre-requisites:

1. Concept of system software, application software, knowledge of input output devices and its usage

Course Objectives:

1. To enable the students to understand basic concepts of operating system.
2. To brief the students about various design aspects of operating system functionality
3. To give hands on exposure to Linux commands and system.

Course Outcomes: After completion of course, students will able to:

1. Recite and Express theoretical and practical aspects of operating system.
2. Infer the concept of process, thread and Inter process communication
3. Outline the concept of deadlocks, necessary conditions for deadlock and various techniques to handle deadlock
4. Analyze memory management policies.
5. Describe file system File and Input /output structure.
6. Infer LINUX and Android Operating system.

UNIT I **OPERATING SYSTEMS OVERVIEW** **(06 Hours)**

Computer System Overview-Basic Elements, Instruction Execution, Interrupts, Multiprocessor and Multicore Organization. Operating system Overview-Kernel, Shell, objectives and functions, Evolution of Operating System- Computer System Organization- Operating System Structure and Operations- System Calls, OS Generation and System Boot., Virtual Machines**.**

UNIT II **PROCESS MANAGEMENT** **(6 Hours)**

Processes-Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication; Threads- Overview, Multicore Programming, Multithreading Models; Thread and SMP Management. Process Synchronization - Critical Section Problem, Mutex Locks, Semaphores, Monitors.

UNIT IV **MEMORY MANAGEMENT** **(6 Hours)**

Main Memory-Contiguous Memory Allocation, Segmentation, Paging, 32 and 64-bit architecture Examples; Virtual Memory- Demand Paging, Page Replacement, Allocation, Thrashing; Allocating Kernel Memory, OS Examples.

UNIT V **INPUT/OUTPUT SYSTEMS** **(6 Hours)**

Mass Storage Structure- Overview, Disk Scheduling and Management; File System Storage-File Concepts, Directory and Disk Structure, Sharing and Protection; File System Implementation- File System Structure, Directory Structure, Allocation Methods, Free Space Management, I/O Systems.

UNIT	CASE STUDY	(6 Hours)
VI	Basic Concepts, of LINUX Multifunction Server, Virtualization- Xen, VMware with Linux Host, Android operating system –Features, characteristics, Basic building blocks, Architecture, System services.	

Term Work:

The sample practical assignments are given below. This can be used as a guideline and course coordinator can recommend the list of practical assignments.

1. Demonstrate the process creation and various states of a process
2. Apply various scheduling algorithms on a process.
3. Apply Banker's algorithm
4. Predict whether a system is in a Safe or Unsafe state.
5. Demonstrate various contiguous memory allocation strategies
6. Demonstrate various page replacement strategies
7. Apply disk Scheduling algorithms

Assignments:

Note:

The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.
For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or
Multiple Choice Questions etc.
The assignments are to be submitted as a hard copy.

Reference Books

1) Dhananjay M Dhamdhere, 'Operating Systems - A Concept Based approach ', Tata McGraw, Hill publication
2 Abraham Silberschatz, Peter B. Galvin &Grege Gagne (Wiley))'. Operating System Concepts '
3) Sumitabha Das, 'Unix Concepts and Applications, Tata McGraw Hill
4)Milan Milinkovic, 'Operating System: Concepts and Design, Tata McGraw Hill
5) Achyut S. Godbole, 'Operating System with case studies in Unix, Netware and Windows NT' Tata McGraw Hill
6) Karim Yoghmour 'Embedded Android', O'Reilly Publication

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

DESIGN AND ANALYSIS OF ALGORITHMS

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks	Theory: 03 Credits
Practical: 2 Hours / Week	Continuous Assessment: 40 Marks	TW/OR: 01Credit
	TW/Oral: 50 Marks	

Course Pre-requisites:

1. The students should possess the knowledge of Data Structures

Course Objectives:

1. To be able to understand the performance analysis of algorithms.
2. To analyze Dynamic Programming Algorithms.
3. Understand NP-Hard, NP- complete Problems.

Course Outcomes: After completion of course, students will able to:

1. Analyze time complexity
2. Analyze space complexity
3. Discuss Divide and Conquer Method
4. Design algorithms using greedy Methods
5. Infer Backtracking
6. Outline NP-Hard and NP-Complete Problems

UNIT I **Introduction**: Algorithm Specification, Pseudocode Conventions, Recursive Algorithms, Performance Analysis: Space Complexity, Time Complexity, Asymptotic Notations, Performance Measurements. **(06 Hours)**

UNIT II **Basic Algorithms**: Basics of Probability Theory, Primality testing, Its Advantages and Disadvantages. Elementary Data Structures: Stack, Queues, Binary Trees, Binary Search trees, Heaps, Heap sort, Graphs, Basic Traversal and Search Techniques. Analysis for complexity of all algorithms. **(06 Hours)**

UNIT III **Divide and Conquer**: General Method, Binary Search, Finding the maximum and minimum, Merge sort, Quick sort, Performance Measurement, Worst case Analysis. Strassens's matrix multiplication. **(06 Hours)**

UNIT IV **Greedy Method**: **(06 Hours)**

The General Method, Knapsack problem, tree vertex splitting, Job sequencing.

Minimum Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Optimal Merge Patterns, Single-Source shortest path. Dynamic Programming: The General Method, Multistage Graph, All pair shortest path, Single Source shortest path, General Weights, Optimal Binary Search Trees, String Editing, 0/1 Knapsack, Traveling Salesman Problem.

UNIT V	**Backtracking:** The General Method, The 8- Queens Problem, Sum of Subsets, Graph Coloring, and Hamiltonian Cycles. Branch and Bound: Least Cost (LC) Search, the 15-puzzle Control abstraction of LC Search, Bounding, FIFO Branch and Bound, LC Branch and Bound.	**(06 Hours)**
UNIT VI	**Algorithm Complexities**: Non deterministic Algorithms, The classes NP-Hard and NP- Complete, Cook's Theorem, NP-Hard Graph Problems, NP-Hard Scheduling Problems, NP-Hard Code Generation Problems. Approximation Problems.	**(06 Hours)**

Term Work:

The sample practical assignments are given below. This can be used as a guideline and course coordinator can recommend the list of practical assignments.

1. Calculate the time complexity of various algorithms.
2. Calculate the space complexity of various algorithms.
3. Implement Knapsack Algorithm.
4. Implement Prim's Algorithm
5. Study and analysis of 8-Queens Problem.
7. Implement Optimal Binary Search Tree.
8. Analyze Quick Sort for Best Case, Worst Case.

Assignments:

Note

The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.

For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.

The assignments are to be submitted as a hard copy.

Text books

1) ElitzHorowith and SartajSahani, S. Rajasekaran, “Fundamentals of Computer Algorithms”, Galgotia Publications.

2) Alfred Aho, John E. Hopcroft, “Design and Analysis of Computer Algorithms”, Pearson Education

References:

1) Thomas Cormen, Charles E Leiserson, Ronald Rivest, “Introduction to Algorithms, Tata Mc-Graw Hill Publication, Second Edition.

2) Rod Stephens, “Essential Algorithms: A Practical Approach to Computer Algorithms”, John Wiley and Sons Publications

3) Jon Kleinberg, Eva Tardos, “Algorithm Design”, Pearson Education

4) Robert Sedgewick, Philippe Flajolet, “An Introduction to the Analysis of Algorithms”, Adison-Wesley Publication, Second Edition

5) Steven S. Skiena, “The Algorithm Design Manual”, Springer Publication, Second Edition.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

DIGITAL SIGNAL PROCESSING

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks	Theory: 03 Credits
Practical: 2 Hours / Week	Continuous Assessment: 40 Marks	TW/OR: 01 Credit
	TW/Oral: 50 Marks	

Course Pre-requisites:

Basic knowledge of Mathematics, Data communication and Microprocessor

Course Objectives:

1. To analyze discrete time signals and systems in time and frequency domain.
2. To represent structures for discrete time systems.
3. To design digital filters and implement using open source software
4. To understand DSP processors and DSP based applications in real world.

Course Outcomes: After completion of course, students will able to:

1. Classify discrete-time signals and discrete time systems.
2. Evaluate LTI system in frequency domain using Fourier Transform, DFT and FFT.
3. Analyze discrete time signals and LTI system using Z transform.
4. Design structures for discrete time systems
5. Design and implement FIR and IIR filters using different methods.
6. Exhibit enhanced architecture and features of DSP processor.
7. Illustrate applications of DSP in speech and Image Processing.

UNIT I **Signals and Systems:** Basic elements of DSP system, Analog to Digital conversion process, Aliasing effect Standard signals, Discrete time systems, classification and properties of discrete time systems, Linear Shift Invariant (LSI) systems, Impulse response, Conditions for causality and stability for LTI systems, Linear convolution. **(06 Hours)**

UNIT II **Fourier Transform** **(06 Hours)**
Fourier transform of standard signals, Discrete Fourier Transform (DFT), DFT of standard signals, properties of DFT, Circular convolution, correlation, Radix-2 FFT algorithms, Decimation in Time (DIT) and Decimation in Frequency (DIF), Inverse DFT and computation of IDFT, relation between ZT, FT and DFT**.**

UNIT III **Z transform** **(06 Hours)**
Region of Convergence (ROC) and its properties, Z transforms of standard signals, properties of Z transform, Inverse Z transform, System functions from Z transform and pole-zero plots, computation of poles and zeros.

UNIT IV **Structures for discrete time systems**: Implementation of general difference equation, Basic structures for FIR systems, Basic structures IIR systems, Representation of structures using signal flow graphs, Feedback in IIR systems, Linear phase FIR filters. **(06 Hours)**

UNIT V **Digital Filters** **(06 Hours)**
Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters, IIR filter

design using impulse invariance and bilinear transformation method, Butterworth approximation, FIR filter design using Windows, Gibb's phenomenon.

UNIT VI **DSP Processors Architecture and applications** **(06 Hours)**

DSP processors fundamentals, Characteristics, Evolution of DSP processors, Comparison of DSP processor and microprocessor, conventional and Enhanced architecture of DSP

Case study: - TMS 320C64X

Applications of DSP in image processing, Feature extraction and pattern matching.

Applications of DSP in speech processing, Speech recognition, speech synthesis, Echo cancellation.

Case study: - Multi-rate signal processing**.**

Term Work:

The sample practical assignments are given below. This can be used as a guideline and course coordinator can recommend the list of practical assignments.

1. WAP to generate samples of Sine, Cosine, Square and Random signal.
2. WAP to compute linear convolution
3. WAP to find N point DFT of a given sequence.
4. WAP to compute Circular convolution
5. WAP to implement Radix-2 DIT FFT Algorithm.
6. WAP to compute Z transform and draw pole zero plot
7. WAP to compute Z transform and draw pole zero plot
8. Find Fourier transform of various window functions
9. Assignment based on DSP applications in Image processing
10. Assignment based on DSP applications in speech processing

Assignment:

Note:

In case of assignments for internal 10 Marks students will be assigned two assignments based on different computer organization and architecture concepts and guided for the respective assignment.

The assignments are to be submitted as a hard copy

Text books

1. John G. Proakis, D.G.Manolakis, "Digital Signal Processing", Pearson Prentice Hall.
2. B.P. Lathi, "Signal Processing and Linear Systems", Oxford University Press.

Reference Books

1. B.Venkataramani, M. Bhaskar, "Digital Signal Processors", Architecture programming & applications, TMH.
2. S.K. Mitra, "Digital Signal Processing Computer Based Approach", TMH.
3. M.H. Hayes "digital signal Processing" Schaum's outlines TMH
4. "TMS 320 C64X CPU & instruction set" Texas instruments reference guide
5. Fredic Harris "Multirate signal processing for communication System", PHI

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

COMPUTER ORGANIZATION AND ARCHITECTURE

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks	Theory: 04 Credits
Tutorial: 1 Hour / Week	Continuous Assessment: 40 Marks	

Course Pre-requisites:

1. Concept of digital logic and microprocessors

2. Basic Understanding of Computer System

Course Objectives:

1. To understand the core concepts about the computer architecture and computer organization.

2. To understand the design of the various functional units of computer system.

Course Outcomes: After completion of course, students will able to:

1. Solve fixed point and floating point arithmetic problems using algorithms.

2. Infer the architecture and functions of Central Processing Unit.

3. Outline the design approaches and functional requirements for implementing control unit.

4. Describe the I/O organization and interconnections

5. Analyze the characteristics of memory system.

6. Infer multiprocessor configuration and modern computer organization.

UNIT I	**CPU structure and function:** Components and functions of computer system, CPU architecture, Processor organization, Register Organization, Instruction Cycle, instruction pipeline. RISC and CISC architecture, The Pentium Processor, Power PC., Superscalar processors.	**(06 Hours)**
UNIT II	**Computer Arithmetic** ALU, Fixed and Floating point numbers, Integer arithmetic, Booth's algorithm, Hardware implementation, Restoring and Non-Restoring Division algorithm, Floating point representation, IEEE standards.	**(06 Hours)**
UNIT III	**Control Unit Organization** Micro-operation and their Register Transfer Language (RTL) specification, Hardware control design methods and implementation, Micro program control, Micro instruction Sequencing, Micro instruction execution.	**(06 Hours)**
Unit IV	**Input/output Organization** Computer system, I/O modules, Programmed I/O, Interrupt driven I/O, Interrupt processing, I/O channels and Processor, DMA, Interface circuits, Bus interconnection, Bus arbitration, Standard buses, Standard interfaces, PCI, SCSI,	**(06 Hours)**

USB bus.

UNIT V	**Memory Organization** Internal memory, Characteristics of memory systems, Memory hierarchy, Error correction, Cache memory organization, Mapping, Replacement algorithms, Pentium cache organization, DDR3 Memory Organization, NUMA and UMA caches. Hard disk drives, RAID levels, Flash drives, USB stick.	**(06 Hours)**
UNIT VI	**Multiprocessor Configuration:** Flynn's classification for multiprocessor system, closely coupled, loosely coupled and tightly coupled configurations. Problems of Bus contentions, Inter Processor communication. Multi core systems, Virtual processors. Case study: - Processor architecture in latest Mobile /Laptop**.**	**(06 Hours)**

Assignment:

Note:

The Term Work prescribed in the syllabus is continuous assessment by the concerned subject faculty.

For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.

The assignments are to be submitted as a hard copy.

Text Books

1. William Stalling, "Computer Organization and Architecture", Fifth edition, PHI
2. Zaky S, Hamacher, "Computer Organization", Fifth Edition, McGraw-Hill Publications, 2001, ISBN 0071122184

Reference books

1. John P Hayes, "Computer Architecture and Organization", McGraw-Hill Publication, 2001, ISBN 0071004793.
2. Carpinelli, "Computer systems Organization & Architecture", Pearson Education India, 2001, ISBN: 8177587676, 9788177587678
3. Morris Mano, "Computer System Architecture, Pearson", Third edition (2008), ISBN: 978-8131700709
4. Andrew S. Tanenbaum "Structured Computer Organization "(5th Edition) ISBN-13: 978-0131485211
5. D.A. Patterson and J.L. Hennessy, "Computer Organization and Design - The Hardware/Software Interface", Morgan KaufmannISBN: 9780124077263

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

ELECTIVE –II: VLSI

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Examination: 60 Marks Continuous Assessment: 40 Marks	Theory:03 Credits

Course Pre-requisites:

1. Analog and Digital Electronics.

2. Semiconductor physics.

Course Objectives:

1. Introduce students to VLSI Design.

2. Introduce students to fabrication and testability techniques.

3. Introduce students to design Low-Power CMOS Logic Circuits.

4. Introduce students to design and simulate digital circuits using FPGA.

Course Outcomes: After completion of course, students will able to:

1. Express concept of Microelectronics and Introduction to MOS Technologies.

2. Outline Layout Design and Tools.

3. Infer Combinational Logical& Sequential Systems.

4. Identify and discuss key problems Dynamic Logic Circuits and its solutions.

5. Relate importance of Low-Power CMOS Logic Circuits.

6. Apply useful criteria for guiding design and evaluation of Chip Input and Output (I/O) Circuits.

UNIT I **Review of Microelectronics and Introduction to MOS Technologies:** MOS, CMOS, BiCMOS Technology. Basic Electrical Properties of MOS, CMOS & BiCMOS Circuits: Ids – Vds relationships, Threshold Voltage VT, Gm, Gds and ωo, Pass Transistor, MOS, CMOS & Bi CMOS Inverters, Zpu/Zpd, MOS Transistor circuit model, Latch-up in CMOS circuits. **(06 Hours)**

UNIT II **Layout Design and Tools:** **(06 Hours)**
Transistor structures, Wires and Vias, Scalable Design rules, Layout Design tools.
Logic Gates & Layouts:
Static Complementary Gates, Switch Logic, Alternative Gate circuits, Low power gates, Resistive and Inductive interconnect delays.

UNIT **Combinational Logic:** **(06 Hours)**
Layouts, Simulation, Delay, Interconnect design, Power optimization, Switch logic and

III Gates.

Sequential Systems:

Memory cells and Arrays, Clocking disciplines, Design, Power optimization, Design validation and testing.

UNIT IV **Dynamic Logic Circuits:** **(06 Hours)**

Introduction, Basic Principles of Pass Transistor Circuits, Voltage Bootstrapping, Synchronous Dynamic, Circuit Techniques, Dynamic CMOS Circuit Techniques, High Performance Dynamic CMOS Circuits, Semiconductor Memories: Introduction, Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), Nonvolatile Memory, Flash Memory, Ferroelectric Random Access Memory (FRAM).

UNIT V **Low-Power CMOS Logic Circuits:** **(06 Hours)**

Introduction, Overview of Power Consumption, Low-Power Design Through Voltage Scaling, Estimation and Optimization of Switching Activity, Reduction of Switched Capacitance, Adiabatic Logic Circuits. BiCMOS Logic Circuits: Introduction, Bipolar Junction Transistor (BJT): Structure and Operation, Dynamic Behavior of BJTs, Basic BiCMOS Circuits: Static Behavior, Switching Delay in BiCMOS Logic Circuits, BiCMOS Applications.

UNIT VI **Chip Input and Output (I/O) Circuits:** **(06 Hours)**

Introduction, ESD Protection, Input Circuits, Output Circuits and L(di/dt) Noise, On-Chip Clock, Generation and Distribution, Latch-Up and Its Prevention. Design for Manufacturability: Introduction, Process Variations, Basic Concepts and Definitions, Design of Experiments and Performance Modelling, Parametric Yield Estimation, Parametric Yield Maximization, Worst-Case Analysis, Performance Variability Minimization. ASIC Design Flow, Introduction to Verilog, Language Constructs and Conventions in Verilog, Gate Level Modeling, Architecture of FPGA.

Assignments:

Note:

1. For internal assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.
2. The assignments are to be submitted as a hard copy.

Reference books

1) Essentials of VLSI Circuits and Systems, K. Eshraghian Eshraghian. D, A. Pucknell, 2005, PHI.

2) Modern VLSI Design – Wayne Wolf, 3rd Ed., 1997, Pearson Education.

3)Introduction to VLSI Systems: A Logic, Circuit and System Perspective – Ming-BO Lin, CRC Press, 2011.

4)Principals of CMOS VLSI Design – N.H.E Weste, K. Eshraghian, 2nd Ed., Addison Wesley.

5)Sung Mo Kang & YosufLeblebici, "CMOS Digital Integrated Circuits: Analysis and Design", Tata McGraw-Hill, Third Edition

6) Neil Weste and K. Eshragian, "Principles of CMOS VLSI Design: A System Perspective", Second Edition,

Pearson Education (Asia) Pvt. Ltd. 2000.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

ELECTIVE –II: NATURAL LANGUAGE PROCESSING

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3Hours / Week	End Semester Examination: 60 Marks Continuous Assessment: 40 Marks	Theory:03 Credits

Course Pre-requisites:

Basic understanding of Theory of Computer Science, Systems Software, Basic mathematics, Probability theory.

Course Objectives:

1. To understand approaches to syntax and semantics in Natural Language Processing and levels of language analysis.
2. To understand current methods for statistical approaches to machine translation.
3. To develop problem solving abilities using Mathematics.

Course Outcomes: After completion of course, students will able to:

1. Outline the Natural Language Processing basics and language representation.
2. Apply various parsing algorithms and its application.
3. Describe language modeling and various clustering techniques.
4. Recite the use of Machine Translation in Natural Language Processing.
5. Apply various filtering methods and semantic networks.
6. Infer advanced tools and parsers for Natural Language Processing.

UNIT I **Introduction to Natural Language Understanding** **(06 Hours)**

The Study of Language Applications of Natural Language Understanding, Evaluating Language Understanding Systems, The Different Levels of Language Analysis, Representations and Understanding the Organization of Natural Language Understanding Systems, Structure of Natural Language Processing (NLP).

UNIT II **Grammars for Natural Language** **(06 Hours)**

Parsing Algorithms, Robust and Scalable Parsing on Noisy Text as in Web documents, Hybrid of Rule Based and Probabilistic Parsing, Human Preferences in Parsing Encoding Uncertainty: Shift-Reduce Parsers, Deterministic Parser Techniques for Efficient Encoding of Ambiguity Partial Parsing, Part of speech tagging.

UNIT III **Natural Language Processing Modeling** **(06 Hours)**

Automatic Morphology Learning, Named Entities, Maximum Entropy Models, Random Fields, Estimation Techniques, and Language Modeling, Parsing and Syntax, The EM Algorithm in Natural Language Processing, Stochastic Tagging, and Log-Linear Models, Probabilistic Similarity Measures and Clustering, Machine Translation, Discourse Processing: Segmentation.

UNIT IV **Natural Language Understanding Methods** **(06 Hours)**

Finite State Machine Based Morphology; Automatic Morphology Learning;

Unsupervised Methods in NLP, Introduction to HMM, HMM Ergodic models, Graphical Models for Sequence Labeling in NLP, Probabilistic parsing, Forward Backward probability, Viterbi Algorithm.

UNIT V	**Ambiguity Resolution** Selectional Restrictions, Semantic Filtering Using Selectional Restrictions, Semantic Networks, Statistical Word Sense Disambiguation, Statistical Semantic Preferences, Combining Approaches to Disambiguation.	**(06 Hours)**
UNIT VI	**Advanced tools, techniques and applications of NLP** Sentiment Analysis; Text Entailment; Robust and Scalable Machine Translation; Cross Lingual Information Retrieval, Some applications like machine translation, database interface, Programming language Python Natural Language Tool Kit (NLTK), NLP applications in web mining and text mining, Parsers for NLP such as Stanford, Open NLP.	**(06 Hours)**

Assignments:

Note:

1. For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.
2. The assignments are to be submitted as a hard copy.

Text books

1) James Allen, "Natural Language Understanding", Pearson Publication, ISBN: 978-81-317-0895-8 2nd Edition
2) D. Jurafsky, J. H. Martin, "Speech and Language Processing", Pearson Education, 2002, 2nd Edition

Reference Books

1) Christopher D. Manning, HinrichSchutze, "Foundations of Statistical Natural Language Processing", 1st edition, Cambridge, Massachusetts, 1999.
2) Tanveer Siddiqui, US Tiwari, "Natural Language Processing and Information Retrieval" Oxford Higher Education.
3) Daniel M. Bikel, ImedZitouni, "Multilingual Natural Language Processing Applications" Pearson Education 1st edition.
4) Lutz and Ascher "Learning Python" O'Reilly ISBN: 0596002815

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

ELECTIVE – II: HUMAN COMPUTER INTERACTION

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Examination: 60 Marks	Theory:03 Credits
	Continuous Assessment: 40 Marks	

Course Pre-requisites:

1. Knowledge of user interface and human psychology.

2. Understanding of Input Output devices.

Course Objectives:

1. To apply knowledge of human psychology, design process on user interface development projects.

2. To provide the user interface designer with concepts and strategies for making design decisions.

3. To expose the user interface designer to tools, techniques and ideas.

4. To identify the importance of good user interface design.

Course Outcomes: After completion of course, students will able to:

1. Express the concept of human computer interaction.

2. Describe the principles of human computer interaction.

3. Outline design goals and standards of HCI designs.

4. Identify and discuss key problems in HCI and its solutions.

5. Recite the importance of software tools and techniques of human factors in developing an interactive system.

6. Apply useful criteria for guiding design and evaluation of user interfaces.

UNIT I **Introduction**: **(06 Hours)**

Human interaction with computers, importance of human characteristics, human consideration, Motivations for Human Factors in design, Eight golden rules of HCI, Murphy's law. Human Psychology and understanding. Case study on human factors.

UNIT II **The graphical user interface**: **(06 Hours)**

Popularity of graphics, the concept of direct manipulation, graphical system, Web user – Interface popularity, Principles of user interface. Current trends in GUI.

UNIT III **Design process:** **(06 Hours)**

Importance of user interface, definition, importance of good design. Benefits of

good design, study of design process components, three Pillars of design, Social Impact for early design review.

UNIT IV	**Software Tools and Techniques in Interface:** Specification methods, interface building tools, Interaction Devices: Keyboard and function keys, pointing devices, speech recognition digitization and generation, image and video displays, pointing devices, navigation schemes.	**(06 Hours)**
UNIT V	**Screen Designing:** Design goals, Screen planning and purpose, organizing screen elements, ordering of screen data and content, screen navigation and flow – visually pleasing composition, presentation information simply and meaningfully information retrieval on web, Case study on screen design. Evaluation of user interface.	**(06 Hours)**
UNIT VI	**Windows &Components:** Selection of window, individual, multiple window design, image browsing and tightly coupled windows. Selection of devices based and screen based controls. Components – text and messages, icons and increases, Multimedia, colors, uses, problems, choosing colors.	**(06 Hours)**

Assignment:

Note:

For internal assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.
The assignments are to be submitted as a hard copy.

Reference books

1) Prece, Rogers, "Sharps Interaction Design", Wiley India.

2)Ben Shneiderman," Designing the user interface". 3rd Edition, Pearson Education Asia

3)Soren Lauesen, "User Interface Design", Pearson Education.

4) Alan Cooper, Robert Reimann, David Cronin, "Essentials of Interaction Design", Wiley.

5)Alan Dix, Janet Fincay, GreGoryd, Abowd, Russell, Bealg," Human Computer Interaction", Pearson Education.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

ELECTIVE – II: DATA STORAGE SYSTEMS

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 3 Hours / Week	End Semester Examination: 60 Marks Continuous Assessment: 40 Marks	Theory:03 Credits

Course Pre-requisites:

In order to understand the content, gain knowledge and successfully complete this course, students should have a basic understanding of the computer architecture, file systems, operating system and networking.

Course Objectives:

1. To enlighten the students with the knowledge of storage systems and to gain exposure to the storage industry

Course Outcomes: After completion of course, students will able to:

1. Describe the data storage systems and its elements
2. Evaluate various types of intelligent storage systems
3. Describe and Evaluate the deployment model of storage systems
4. Evaluate the various storage networking technologies
5. Infer the key processes for managing the storage infrastructure
6. Determine the appropriate storage solution for a given scenario

UNIT I **Introduction to Data Storage** **(06 Hours)**

Data, Data Variety, Information, knowledge, Big Data, Data explosion, Storage, **Storage devices and its types**, Memory hierarchy, Secondary storage, tertiary storage. Storage architecture and its evolution, Data center, Components of Data center, Managing Data center.

Introduction to Application workloads, DBMS, Compute, Memory Virtualization, Device Driver, Volume Manager, File System, Object storage, Block storage, Server Virtualization.

UNIT III **File System and Storage System** **(06 Hours)**

Local file system, Journaling, Snapshots, Network file systems, Principle, NAS, SAN, DAS.

Case Study: DAFS, Shared Disk File System, GPFS, Comparison of FC SAN, FCoE SAN, iSCSI SAN and NAS. Scale up and scale out architectures.

Protocols: SCSI, iSCSI, SC,FCoE, CIFS vs NFS

UNIT IV **I/O path and Storage Virtualization** **(06 Hours)**

Define Virtualization, define storage virtualization, I/O path, Physical channel, Elements in Physical I/O Path, software Layers of I/O Path, Software stack, I/O path virtualized, Demerits and need of storage virtualization, Virtualization entity, replaceable storage device, use of dynamic storage allocation, use of data migration, virtualization on block and file level, virtualization at network level, symmetric and asymmetric.

UNIT V **Function and Network technologies** **(06 Hours)**

Functions: Instant Copies, Remote Mirroring, consistency groups, LUN Masking, Availability of Disk Storage systems.

Storage network: Transmission technique and protocol, SCSI and storage network, Fiber channel, Link, ports and Topology, IP storage, InfiniBand.

UNIT VI **Performance Monitoring and Management** **(06 Hours)**

Performance management: Why analyze performance, Capacity Management: capacity planning, I/O Capacity, SLA's, Reactive Style, Casually Observant, Actively Observant, Proactive style, Performance Lifecycle, Performance Hierarchy.

Performance Metrics: IOPS, part of an I/O, Throughput, Latency, Utilization, understanding application workload.

Tools: Linux performance commands: TOP, IOSTAT, SAR, IOMeter and Iozone.

Assignments:

Note

For internal Assessment of 10 Marks, students have to submit two assignments based on problems of different types of any programming assignment or theory assignment or any case study or quiz or Multiple Choice Questions etc.

The assignments are to be submitted as a hard copy.

Reference Books

1) Information Storage and Management, second edition, EMC education Services, John Wiley & Sons, Inc.

2)Storage Networks Explained, second edition, Ulf Troppens, Rainer erkens, Wolafka, haustein, Wolfgang, Wiley, SNIA.

3)Introduction to storage area networks and system networking, An IBM Redbooks Publication, Jon Tate, Pall Beck, Hector Hugo Ibarra, Shanmuganathan Kumaravel, Libor Miklas

4)The Design and Implementation of a Robust-storage-system Architecture, Robert C. Good, University of Waterloo, 1995,

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

PROGRAMMING LAB- IV

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Practical: 4 Hours / Week	Term Work (Practical): 50 Marks	Credits: 02 Credits

Course Pre-requisites:

1. Students are expected to have some rudimentary knowledge of computer networking in general and a working knowledge of system programming in C/C++ under Unix/Linux.

Course Objectives:

1. To understand inter-process and inter-system communication.
2. To understand socket programming in its entirety.
3. To understand usage of TCP/UDP / Raw sockets.

Course Outcomes: After completion of course, students will able to:

1. Outline the fundamentals of Unix environment.
2. State Process handling.
3. Recite Inter-Process Communication
4. Implement Unix Socket Programming.
5. Infer fundamental concepts of Shell Programming.
6. Implementation of shell meta-characters.

UNIT I **Overview of Unix OS** **(06 Hours)**

UNIX Architecture, UNIX Standardization, Files and Directories, Unix Installation, Input and Output, Programs and Processes, Error Handling, Logging in, User Identification, System Calls and Library Functions, Signals & Time Values

UNIT II **Process Environment** **(06 Hours)**

Process Identifiers, Fork Function, Vfork Function, Exit Functions, Wait and Waitpid Functions, Waitid Function, Wait3 and Wait4 Functions, Process Accounting, User Identification, Terminal Logins, Network Logins, Process Groups, Sessions, Introduction of Daemon Processes. Performance Monitoring Tools: Process Status (ps), Top, Xosview, Treeps.

UNIT III **Inter-process Communication** **(06 Hours)**

System V IPC. Message Passing: Pipes and FlFOs, Message Queues: System V, Synchronization: Mutexes and Condition Variables, Read-Write Locks, Record Locking, System V Semaphores, System V Shared memory.

UNIT IV **Network IPC: Sockets** **(06 Hours)**

Introduction, Socket Descriptors, Addressing, Connection Establishment, Transport Layer, Socket Introduction, TCP Sockets, UDP Sockets, Raw Sockets, Socket Options, I/O Multiplexing, Name and Address Conversions.

UNIT V **Introduction to Unix Shell Programming** **(06 Hours)**

Types of Shells, Interactive Shell Scripts, Shell variables, Shell keywords, Positional Parameters, Command line arguments, shell script Arithmetic, Control Instructions, Loop control structure.

Shell Metacharacters: Filename Substitution Metacharacters, I/O Redirection,

Process execution, Conditional execution using && and ||, Quoting metacharacters, Special Parameters, Debugging a Script $* and S@

UNIT VI **Unix Network Tools** **(06 Hours)**

Unix Network Commands related to Connectivity, network interface commands, Routing, Arp, NFS/NIS etc.
Windows power shell: purpose of PowerShell, Windows management framework, cmdlets and understanding syntax.

Term Work:

The sample practical assignments are given below. This can be used as a guideline and course coordinator can recommend the list of practical assignments.

1. Introduction to Unix commands using vi editor.
2. Implementation of Unix System calls
3. Write a program to implement message passing using Pipes and FIFOs.
4. Write a program to implement synchronization using semaphores.

5 Write a program to implement synchronization using Mutex variable.

6 Write a program to implement client-server communication using TCP sockets.

7. Write a program to implement client-server communication using UDP sockets.
8. Write a program to implement client-server communication using Raw sockets.
9. Write a shell script to pass command line arguments.
10. Write a shell script using metacharacters for I/O redirection.

Reference books

1. W. Richard Stevens, Stephen A. Rago, "Advanced Programming in Unix Environment",3rd Edition, Addison-Wesley Professional.
2. W. Richard Stevens "UNIX Network Programming-Interprocess communication", Volume 2, Second Edition, Prentice Hall Publication.
3. "UNIX Network Programming: The Sockets Networking Api", Volume 1, 3rd Edition, Addison-Wesley Professional Computing.