



# **Fergusson College (Autonomous) Pune**

Learning Outcomes-Based Curriculum

for 1 or 2 years M.Sc. Programme

as per guidelines of

**NEP-2020**

**M.Sc. I - Computer Applications**

With effect from June 2024

<b>Program Outcomes (POs) of M.Sc. Computer Applications</b>	
<b>PO1</b>	<b>Disciplinary Knowledge:</b> Demonstrate comprehensive knowledge of the discipline that forms a part of a postgraduate programme. Execute strong theoretical and practical understanding generated from the specific programme in the area of work.
<b>PO2</b>	<b>Critical Thinking and Problem solving:</b> Exhibit the skill of critical thinking and understand scientific texts and place scientific statements and themes in contexts and also evaluate them in terms of generic conventions. Identify the problem by observing the situation closely, take actions and apply lateral thinking and analytical skills to design the solutions.
<b>PO3</b>	<b>Social competence:</b> Exhibit thoughts and ideas effectively in writing and orally; communicate with others using appropriate media, build effective interactive and presenting skills to meet global competencies. Elicit views of others, present complex information in a clear and concise and help reach conclusion in group settings.
<b>PO4</b>	<b>Research-related skills and Scientific temper:</b> Infer scientific literature, build sense of enquiry and able to formulate, test, analyse, interpret and establish hypothesis and research questions; and to identify and consult relevant sources to find answers. Plan and write a research paper/project while emphasizing on academics and research ethics, scientific conduct and creating awareness about intellectual property rights and issues of plagiarism.
<b>PO5</b>	<b>Trans-disciplinary knowledge:</b> Create new conceptual, theoretical and methodological understanding that integrates and transcends beyond discipline-specific approaches to address a common problem.
<b>PO6</b>	<b>Personal and professional competence:</b> Perform independently and also collaboratively as a part of a team to meet defined objectives and carry out work across interdisciplinary fields. Execute interpersonal relationships, self-motivation and adaptability skills and commit to professional ethics.
<b>PO7</b>	<b>Effective Citizenship and Ethics:</b> Demonstrate empathetic social concern and equity centered national development, and ability to act with an informed awareness of moral and ethical issues and commit to professional ethics and responsibility.
<b>PO8</b>	<b>Environment and Sustainability:</b> Understand the impact of the scientific solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development.
<b>PO9</b>	<b>Self-directed and Life-long learning:</b> Acquire the ability to engage in independent and life-long learning in the broadest context of socio-technological changes.

<b>Program Specific Outcomes (PSOs) for M.Sc. Computer Applications Program</b>	
<b>PSO1</b>	<b>Academic competence:</b> Identify, formulate and solve real world computing system problems of various domains by understanding and applying the principles Of mathematics, computing techniques and business concepts.
<b>PSO2</b>	<b>Personal and Professional Competence:</b> Analyse, design, test, develop and maintain desktop, web, mobile and cross platform software applications using modern tools and technologies in economical and affordable way.
<b>PSO3</b>	<b>Research Competence:</b> Identify the potentiality to design experiments for complex problems, collect data, interpret data, develop, implement computer programs and use knowledge in various domains to identify research gaps and hence to provide solutions to new ideas and innovations.
<b>PSO4</b>	<b>Entrepreneurial and Social competence:</b> Analyse and understand the evolutionary changes in computing by inculcating employability and entrepreneur skills among the lifelong learners with moral values and ethics. Develop skills required for social interaction.

**Fergusson College (Autonomous), Pune**  
**Proposed First Year Curriculum as per NEP 2020**  
**Department of Computer Science**

**M.Sc. Computer Applications**

Semester	Paper Code	Paper Title	Type	Credits
<b>I</b>	CSA -501	Software Engineering Concepts	Theory	4
	CSA -502	Operating Systems	Theory	4
	CSA -503 <b>OR</b>	Mathematical Foundation	Theory	<b>4</b>
	CSA -504	<b>Or</b> Programming Paradigms	Theory	
	CSA -510	Research Methodology	Theory	4
	CSA -520	Practical - I	Practical	2
	CSA -521	Practical - II	Practical	2
	<b>Total Semester Credits</b>			
<b>II</b>	CSA -551	Networking Concepts	Theory	4
	CSA -552	Algorithms and Data Structure	Theory	4
	CSA -553 <b>OR</b>	Java Programming	Theory	<b>4</b>
	CSA -554	Dot Net Framework	Theory	
	CSA -560	The Art of Software Crafting	Project	4
	CSA -570	Practical - III	Practical	2
	CSA -571	Practical - IV	Practical	2
	<b>Total Semester Credits</b>			
<b>Total PG-I Credits</b>				<b>40</b>

**Teaching and Evaluation (Only for FORMAL education courses)**

Course Credits	No. of Hours per Semester Theory/Practical	No. of Hours per Week Theory / Practical	Maximum Marks	CE 40 %	ESE 60%
1	15 / 30	1 / 2	25	10	15
2	30 / 60	2 / 4	50	20	30
3	45 / 90	3 / 6	75	30	45
4	60 / 120	4 / 8	100	40	60

**Eligibility: As per the rules and regulations of Savitribai Phule Pune University (SPPU)**

<b>F.Y. M.Sc. Semester I</b>		
<b>CSA-501</b>	<b>Software Engineering Concepts</b>	<b>Credits: 04</b> <b>Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Identify the factors that direct Software Project and outline the role of process model in a Project.	
CO2	Exemplify, SRS to develop use cases for the project.	
CO3	Carry out life cycle activities like Analysis, Design, Implementation, Testing and Maintenance. Apply the knowledge, techniques, and skills in the development of a software product.	
CO4	Appraise, various processes used in all the phases of the product	
CO5	Evaluate the background and driving strengths for taking an Agile approach to Software Development	
CO6	Design real time applications using Object Oriented Analysis and Design Methodology.	

<b>Unit No.</b>	<b>Contents</b>	<b>No. of Hours</b>
<b>1</b>	<p><b>SOFTWARE PROCESS</b></p> <p>Process models – Defining a Framework Activity, Process Patterns, Process Assessment and improvement - Prescriptive Process Models – Specialized process models- The Unified Process – Personal and Team Process models – Process Technology – Product and Process.</p> <p><b>Practical Component</b></p> <p>1. Explore and Practice different types of case tools such as Rational Rose / other Open Source</p>	<b>10</b>
<b>2</b>	<p><b>UNDERSTANDING REQUIREMENTS</b></p> <p>Requirements Engineering – Eliciting requirements – Developing use cases – Building the requirement model – Negotiating and validating requirements – Scenario Based Modelling – UML Models – Data modelling concepts– Patterns for Requirement modelling.</p> <p><b>Practical Component</b></p> <p>1. Apply Project Planning and Software Requirement Analysis to at least one application problem.</p>	<b>12</b>

<b>3</b>	<p><b>DESIGN CONCEPTS</b></p> <p>Design Process – Design concepts – Software Architecture – Architectural Styles and Design – Assessing alternative architectural designs – architectural Mapping Using Data Flow – Component Level Design – Designing Class Based Components – Component level design for Web Apps – Designing Traditional Components – User Interface Design.</p> <p><b>Practical Component</b></p> <p>Apply the following to typical application problems: a. Software Design b. Data Modelling &amp; Implementation, Software Estimation</p>	<b>14</b>
<b>4</b>	<p><b>Software testing</b></p> <p>Test Strategies for Conventional Software – OO Software and testing – Validation testing – System Testing – The art of debugging – Internal and External views of testing – Basis path testing –White Box testing – Control structure testing – Block Box Testing – Model based Testing – Patterns for Software Testing.</p> <p><b>Practical Component</b></p> <p>Apply the software Testing concept for any one possible application.</p>	<b>12</b>
<b>5</b>	<p><b>AGILE METHODOLOGY AND SOFTWARE PROCESS</b></p> <p>What is agility – Agility and cost of change – What is an agile process – Extreme programming – Agile Process models – Tool set for the agile process – Software Process Improvement – SPI Process – CMMI – People of CMM – SPI Framework – SPI Return on Investment – SPI Trends.</p> <p><b>Practical Component</b></p> <p>Apply Software Quality Checking to at least one of the applications.</p>	<b>12</b>

#### **Learning Resources:**

1. Roger S Pressman, “Software Engineering”, Tata McGraw- Hill Publications
2. I. Sommerville, “Software Engineering”, Addison Wesley
3. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, PHI
4. R. Fairley, Software Engineering Concepts, Tata McGraw-Hill Edition.

#### **E- BOOKS**

1. <http://www.ddegjust.ac.in/studymaterial/mca-3/ms-12.pdf>

#### **MOOC**

1. <https://www.classcentral.com/course/swayam-software-engineering-14293>
2. <https://archive.nptel.ac.in/courses/106/101/106101061/>

## F.Y. M.Sc. Semester I

<b>CSA-502</b>	<b>Operating Systems</b>	<b>Credits: 04</b> <b>Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Describe the services and system calls provided by the operating system.	
CO2	Discuss and articulate classical problems of process synchronization.	
CO3	Apply process management and memory management concepts to solve different software problems.	
CO4	Identify and analyse deadlock handling situations.	
CO5	Evaluate the performance of different Disk scheduling algorithms and Page replacement algorithms.	
CO6	Propose the best CPU Scheduling algorithm for a given problem instance.	
Unit No.	Contents	No. of Hours
<b>1</b>	<b>Pre-operating system concepts</b> CPU, Mother Board, Different types of registers, RAM, Types of RAMS, Hard Disk, Types of Hard Disk, Hard Disk Partitions, Types of HD Partition, what is Boot Sector	<b>4</b>
<b>2</b>	<b>Booting in Operating System</b> Firmware, Sequencing of Booting, Boot loader, Types of booting, BOIS, Boot process, what is Dual Booting?	<b>4</b>
<b>3</b>	<b>Introduction to Operating System</b> Definition of operating system, kernel, types of kernels, comparison between Multiprogramming and Time-sharing operating system, Difference between CUI and GUI.	<b>3</b>
<b>4</b>	<b>Process Management</b> Process definition, types, process execution mode, Process state transition, Process Control Block, Process scheduling, Time related to the process, Scheduling Algorithms, Types of schedulers, Context Switch. What is thread? Types of thread.	<b>12</b>
<b>5</b>	<b>Process Synchronization</b> Introduction, types of process, IPC, Race Condition Critical section problem, Semaphores: Concept, Implementation, Deadlock & Starvation, Classic Problems of synchronization	<b>10</b>

<b>6</b>	<b>Memory Management</b> Introduction to memory management, Fix partition, Dynamic Partition, Logical vs. physical addresses, Dynamic vs. Static linking, fragmentation Swapping, Paging, Segmentation, Virtual memory, Demand paging.	<b>9</b>
<b>7</b>	<b>Information Management</b> File, Attributes of the File, Operations on the File, file allocation Method, free space management, what is file systems, types of file system., Disk Scheduling Algorithm	<b>6</b>
<b>8</b>	<b>Advance Concepts in Operating system</b> Multiprocessing Operating system, System Calls in Operating System, Distributed Operating System, Multithreading Models in Operating system, Embedded Operating System, Distributed File System, Difference between 32 bit and 64-bit Operating System, Device Driver in Operating System.	<b>8</b>
<b>9</b>	<b>Case Study</b> Introduction, Features of Linux, how to manage your Linux, Linux commands or Installation of Windows, windows command	<b>4</b>

#### **Learning Resources:**

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating Systems, – Willey Publication
2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating Systems, – Willey Publication
3. Pabitra Pal Choudhary, Operating Systems: Principles and Design – (PHI Learning Pvt. Ltd), Paperback.
4. William Stallings, Operating Systems: Internals and Design Principles, Prentice Hall (7<sup>th</sup> Edition), 2017

#### **Web Resources:**

1. <https://www.cse.iitb.ac.in/~mythili/os/>
2. <https://www.youtube.com/watch?v=mXw9ruZaxzQ>
3. <https://archive.nptel.ac.in/courses/106/105/106105214/>



## F.Y. M.Sc. Semester I

<b>CSA-503</b>	<b>Mathematical Foundation</b>	<b>Credits: 04</b> <b>Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Describe fundamentals of set theory, relations, functions and graph	
CO2	Explain different concepts of statistics using R programming.	
CO3	Apply the concepts of graphs to solve various problems in day-to-day life.	
CO4	Analyse various measures of location and dispersion on real data.	
CO5	Evaluate the fundamental probability concepts.	
CO6	Generate a Jordan canonical form by applying concepts of eigen values and eigen vectors	

Unit No.	Contents	No. of Hours
<b>1</b>	<b>Relation:</b> Relations, Properties of Binary relation, Matrix representation of relations, Equivalence relations	<b>4</b>
<b>2</b>	<b>Linear algebra</b> Elementary transformations of a matrix. Rank of a matrix by elementary row transformations, Consistency of a system of linear nonhomogeneous algebraic equations. Eigen values and Eigen vectors of a square matrix, Diagonalization of a square matrix of order 3	<b>8</b>
<b>3</b>	<b>Recurrence Relations</b> , Solving Linear Recurrence Relations, Generating Functions	<b>8</b>
<b>4</b>	<b>Graphs:</b> Introduction to graphs, Directed and undirected graph, Isomorphic graphs, Subgraph, Cut Points and Bridges, Weighted graph, paths and circuits, shortest path in weighted graphs, Eulerian graph and Hamiltonian graph, planar graphs. Euler's Formula.	<b>16</b>
<b>5</b>	<b>Statistics (using Excel):</b> Measures of Location, Measures of Dispersion, Measures of Skewness and Kurtosis, Quantiles and Box plot, Linear Models – Simple Linear Regression, Multiple Linear Regression	<b>12</b>
<b>6</b>	<b>Probability and Probability Distributions:</b> Introduction to Probability, Conditional Probability, Baye's Theorem, Sensitivity and Specificity Analysis, Concepts of discrete and continuous probability distribution, Binomial Distribution, Poisson Distribution, Normal Distribution	<b>12</b>

**Learning Resources:**

1. J. L. Mott, A. Kandel, T. P. Baker, Discrete Mathematics for Computer Scientists and Mathematicians: PHI.
2. John Truss, Discrete Mathematics for Computer Science: Pearson International, 2001.
3. Introduction to Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill.
4. Liu, Computer Science: Mathematical Introduction: PHI.
5. Statistical Methods, G.W. Snedecor, W.G. Cochran, John Wiley & sons, 1989.
6. A Beginners Guide to R, Alain Zuur, Elena Leno, Erik Meesters, Springer, 2009
7. Statistics Using R, Sudha Purohit, S.D.Gore, Shailaja Deshmukh, Narosa, Publishing Company
8. Introduction to Linear Regression Analysis, Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining, Wiley
9. <https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/>

<b>F.Y. M.Sc. Semester I</b>		
<b>CSA-504</b>	<b>Programming Paradigm</b>	<b>Credits: 04</b> <b>Hours: 60</b>
<b>Course Outcome (COs)</b>		
<b>On completion of the course, the students will be able to:</b>		
CO1	Identify different programming paradigm and their use cases.	
CO2	Summarize the different tools and understand their use	
CO3	Understand and apply the concepts that form the basis of functional, logic and concurrent programming paradigms	
CO4	Analyse the different data types with respective their needs	
CO5	Justify the use of run time program management and late binding	
CO6	Formulate abstractions with procedures and data in different programming paradigms.	

<b>Unit No</b>	<b>Contents</b>	<b>No. of Hours</b>
<b>1</b>	<p><b>Introduction</b></p> <p><b>Overview</b> the main programming paradigms: imperative vs declarative, structured, object-oriented, concurrent, functional, Scripting languages, Data-oriented languages, event-driven programming</p> <p><b>Standardization</b> Purpose of standardisation, Benefits for developers and tool vendors, Potential drawbacks of standardisation, standardisation bodies, e.g., ECMA, ANSI, ISO</p>	<b>6</b>
<b>2</b>	<p><b>Programming environments</b></p> <p>Process of converting code into executable programs, Compilation, Interpreters, Hybrid use of compilation and interpreters, Steps for compilation and interpreters, e.g.: Tokenising, Parsing, Code generation, Linker, loader</p> <p><b>Tools</b></p> <p>Interactive development tools: purpose and architecture framework of IDE, common features like Code editing, Syntax highlighting, Code completion</p> <p><b>Debugging tools:</b> Purpose of debugging in development, features like run, Pause, Step into, Step over, Inspecting values, Watch values, Breakpoints</p> <p><b>Testing tools:</b> Purpose of testing tools, Automated testing, e.g.: Using unit testing frameworks, UI testing frameworks, Use of static checking, e.g.: Lint</p> <p>Role of testing in continuous integration</p>	<b>12</b>

	<b>Configuration management:</b> Build tools, Version control systems, e.g.: Git, Subversion, Building different versions of software	
<b>3</b>	<b>Names, Scopes and Bindings</b> Names and Scopes, Binding Time, Scope Rules, Storage Management, Binding of Referencing Environments. <b>Control Flow:</b> - Expression Evaluation, Structured and Unstructured Flow, Sequencing, Selection, Iteration, Recursion, Non-determinacy.	<b>14</b>
<b>4</b>	<b>Data Types:</b> -Type Systems, Type Checking, Records and Variants, Arrays, Strings, Sets, Pointers and Recursive Types, Lists, Files	<b>9</b>
<b>5</b>	<b>Subroutines and Control Abstraction:</b> - Static and Dynamic Links, Calling Sequences, Parameter Passing, Generic Subroutines and Modules, Exception Handling, Co-routines.	<b>9</b>
<b>6</b>	<b>Concurrency:</b> - Threads, Synchronization. Run-time program Management: - Virtual Machines, Late Binding of Machine Code, Reflection, Symbolic Debugging, Performance Analysis	<b>10</b>

### Learning Resources:

1. Concepts of Programming Languages, Sebesta, R., Pearson
2. Programming Language Pragmatics, Scott, M., Morgan Kaufmann
3. Programming Languages: Principles and Paradigms, Gabbrielli, M., Martini, S., Springer
4. Concepts in Programming Languages, J. C. Mitchell, Cambridge University Press
5. Programming Languages: Principles and Paradigms, A. Tucker and R. Noonan, McGraw-Hill

### E- Book

1. <https://cslab.pepperdine.edu/warford/cosc450/>

<b>F.Y. M.Sc. Semester I</b>		
<b>CSA-510</b>	<b>Research Methodology</b>	<b>Credits: 04</b> <b>Hours: 60</b>

**Instructions:**

- A student will choose 4 suitable modules depending on his/her faculty. The contact hours will be around 60 based on assignments and examinations. The examination for each module will be separately performed.

**Mode of examination-**

- The internal examination for each module will be separately conducted. The examination mode is decided by the instructor of that module.
- The external examination will be conducted at the time of 4th half yearly progress review. The student's implementation of various aspects in research methodologies will be checked.

<b>Unit No.</b>	<b>Contents</b>	<b>No. of Hours</b>
<b>1</b>	History of research. Indian, Egyptian, Greek ideas methodologies and research in agriculture, chemistry, metallurgy, medical. Ancient Indian research methodology applications.	6
<b>2</b>	Statistical analysis and its significance. , Exploratory and confirmatory research, Planned and ad-hoc methods of data collection, non-response and methods of recovering the missing response, Various software for statistical analysis. The module will consist of case studies of the research performed in various subjects using statistical methods, Error and noise analysis, curve fitting.	12
<b>3</b>	Creating questionnaire. Data analysis from answers, Selection of research topic (case study based). Selection of research topic (case study based)	8
<b>4</b>	Literature search, selection of research topic (case study based), maintaining laboratory records (case study based). Safety in Laboratories, Ethical considerations, effective verbal and non-verbal communication, field data collection, safety in field.	12
<b>5</b>	Writing research paper and/or thesis, making a presentation, writing a research proposal, and patents in Science, technology.	11
<b>6</b>	<b>Databases and Research Metrics</b> Databases (i) Indexing databases. (ii) Citation databases: Web of Science, Scopus, etc. Research Metrics (i) Impact Factor of journal as per Journal Citation Report, SNIP, SJR, IPP, Cite Score. (ii) Metrics: h-index, g index, i10 index, altimetry	5

<b>7</b>	<p><b>Research Methods for Computer Science</b></p> <p>Formal Methods: Formal Specification, Algorithm, and Complexity; Building Artefacts: Proof of Performance, Proof of Concept, and Proof of Existence; Process Methodology: Methods for Software Engineering and Human-Computer Interaction, Cognitive Processes, Interactive Games, Social Networks, and Web Analytics</p>	<b>6</b>
----------	--	----------

**Learning Resources:**

1. 'History of the Scientific Methods' by Martin Shuttleworth, <https://explorable.com/history-of-the-scientific-method>.
2. The Statistical Analysis of Experimental Data' by, John Mandel, ISBN: 0486646661, ISBN13: 9780486646664

<b>F.Y. M.Sc. Semester I</b>		
<b>CSA-520</b>	<b>Programming Concepts Implementation (Experiential learning)</b>	<b>Credits: 02 Hours:60</b>

<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>	
CO1	Understand the programming language and create an awareness about programming paradigm
CO2	To provide exposure to problem-solving through C programming.
CO3	Develop programs using the basic elements like control statements, Arrays and Strings
CO4	Apply code reusability with functions and pointers
CO5	Implement different Operations on arrays, functions, pointers, structures, and unions.
CO6	Understand the basics of file handling mechanisms

<b>Unit No.</b>	<b>Contents</b>
<b>1</b>	<b>Introduction to Programming</b> Basic Difference between Procedure Oriented Language and Object-oriented Language, Concepts of Machine level, Assembly level and High-level programming, Flow charts and Algorithms
<b>2</b>	<b>Fundamentals of 'C'</b> Features of C language, structure of C program, comments, header files, data types, constants and variables, operators, expressions, evaluation of expressions, type conversion, precedence and associativity, I/O functions.
<b>3</b>	<b>Control Structures in 'C'</b> Simple statements, Decision making statements, Looping statements, Nesting of control structures, break and continue statement, goto statement
<b>4</b>	<b>Array &amp; String</b> Concept of array, One- and Two-dimensional arrays, declaration and initialization of arrays, String, String storage, Built-in string functions
<b>5</b>	<b>Functions</b> Concept of user defined functions, prototype, definition of function, parameters, parameter passing, calling a function, Macros, Pre-processing
<b>6</b>	<b>Pointers</b> Basics of pointers, pointer to pointer, pointer and array, pointer to array, array of pointers, function returning a pointer

<b>7</b>	<b>Structure and Union</b> Basics of structure, structure members, accessing structure members, nested structures, array of structures, structure and functions, structures and pointers, unions, bit-fields
<b>8</b>	<b>File Management</b> Introduction to file management, Simple file management functions for text files, reading from and writing to files

### **Learning Resources**

1. Let us C, Yashwant Kanitkar
2. C: The Complete Reference, Herbert Schildt, McGrawHill
3. Computer fundamentals and Programming in C, Pradip dey and Manas Ghosh, Oxford
4. Brian W. Kernighan and Dennis M. Ritchie: The C Programming Language, Prentice Hall Publication

### **Web References:**

1. <http://www.programmingsimplified.com/c-program-examples>
2. [http://en.wikipedia.org/wiki/C\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_%28programming_language%29)



<b>F.Y. M.Sc. Semester I</b>		
<b>CSA-521</b>	<b>Database Management (Experiential Learning)</b>	<b>Credits: 02 Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Describe the basic concepts of database management systems.	
CO2	Discuss the concepts of concurrency control, transaction processing and recovery management.	
CO3	Solve queries using SQL	
CO4	Analyse database requirements and determine the entities involved in the system and their relationship.	
CO5	Evaluate the normality of a logical data model and correct anomalies.	
CO6	Develop programs in PL/SQL on Stored functions, Cursors, Triggers and Views	

<b>Unit No</b>	<b>Contents</b>
<b>1</b>	<p><b>Introduction to Databases:</b> Introduction, Characteristics of database approach, Advantages of using the DBMS approach, Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment</p> <p><b>Conceptual data modelling:</b> Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization.</p>
<b>2</b>	<p><b>An Introduction to RDBMS:</b> Relational Database Management System, RDBMS Properties, Maintaining Integrity and Defining Data Integrity, Integrity Rules and Integrity Constraints, Relational Integrity Rules.</p>
<b>3</b>	<p><b>SQL</b> Types of SQL, DCL- DML, Basic queries in SQL Single table, Deletion- Insertion- and Update in SQL, Simple queries (with insert, delete, and update), Multi table Retrievals, Joins, Nested queries (with foreign key and using multi tables)</p>
<b>4</b>	<p><b>Stored Functions</b> Function definition, how to write function and its execution, Solving some problems with function</p>
<b>5</b>	<p><b>Stored Procedures</b> Procedure definition, how to write procedure and its execution, Solving some problems with procedure</p>

<b>6</b>	<b>Cursors</b> Cursor definition, how to write cursor and its execution, Solving some problems with cursor
<b>7</b>	<b>Triggers</b> Trigger definition, how to write trigger and its execution, Solving some problems with trigger
<b>8</b>	<b>Views</b> View definition, how to write view and its execution, Solving some problems with view
<b>9</b>	<b>Introduction to other database</b> Distributed Databases: Architecture, Data Storage, Transaction Processing Object-based Databases: Object Database Concepts, Object-Relational features, ODL Spatial databases-Type of spatial data–Indexing in spatial databases, Mobile Databases– Transaction Model in MDS XML Databases: XML Hierarchical Model, DTD, XML Schema, XQuery Information Retrieval: IR Concepts, Retrieval Models, Queries in IR systems. Introduction to NoSQL Databases: Main characteristics of Key-value DB, Document DB, Main characteristics of Column – Family DB and Graph DB

### **Learning Resources:**

1. Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education
2. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGrawHill
3. R Ramakrishnan, J Gehrke, Database Management Systems, McGraw-Hill,
4. Peter Rob & Carlos Coronel, Database Systems design, Implementation, and Management, Cengage Learning
5. Thomas M. Connolly and Carolyn Begg “Database Systems: A Practical Approach to Design, Implementation, and Management”, 6th edition, Pearson India
6. Thomas Nield, Getting Started with SQL, O’Reiley
7. John Viescas, SQL Queries for Mere Mortals, Addison-Wesley
8. James Groff, SQL: The Complete Reference, McGraw-Hill Education
9. Christopher D.manning, Prabhakar Raghavan, Hinrich Schutze, An introduction to Information Retrieval, Cambridge University Press

### **E- BOOKS**

1. [https://mrcet.com/downloads/digital\\_notes/CSE/II%20Year/DBMS.pdf](https://mrcet.com/downloads/digital_notes/CSE/II%20Year/DBMS.pdf)
2. [https://www.bput.ac.in/lecture-notes-download.php?file=lecture\\_note\\_420507181042070.pdf](https://www.bput.ac.in/lecture-notes-download.php?file=lecture_note_420507181042070.pdf)

<b>F.Y. M.Sc. Semester II</b>		
<b>CSA-551</b>	<b>Networking Concepts</b>	<b>Credits: 04 Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Define different data communication components in networking and describe its diverse types of models and topologies.	
CO2	Explain Random access, controlled access and channelization protocols and discuss design issues.	
CO3	Apply different line coding, transmission methods, switching techniques and examine their functions, error control techniques and protocols.	
CO4	Analyse different network models and addressing schemes.	
CO5	Review different architectures related to different protocols at different layers.	
CO6	Design subnet mask and classful IP addresses.	

<b>Unit No.</b>	<b>Contents</b>	<b>No. of Hours</b>
<b>1</b>	<b>Introduction to Computer Networks</b> Data Communication: Characteristics of data communication, Components, Data representation, Data flow Computer Networks: Distributed processing, Physical Structure-Point to Point, Broadcast, Categories of topology (mesh, star, ring, bus, etc.) Categories of network: LAN, WAN, MAN, INTERNET etc. Protocols and Standards: Definition of protocol, Key elements, Defacto & Dejure standard, Standards organizations Network Software: Protocol Hierarchies –layers, protocols, peers, interfaces, network architecture - protocol stack, design issues of the layers - addressing, error control, flow control, multiplexing and de-multiplexing, routing, Connection-oriented and connectionless service, Service Primitives listen, connect, receive, send, disconnect, The relationships of services to Protocol	<b>6</b>
<b>2</b>	<b>Network Models</b> OSI Reference model- Functionality of each layer TCP/IP model - Introduction to IP, TCP & UDP, TCP/IP Protocol Suite Addressing - Physical, Logical & Port addresses	<b>4</b>
<b>3</b>	<b>The Physical Layer</b> The Basic Concepts of analog & digital data and signals Line Coding digital to digital conversion: Characteristics, Line Coding Schemes: Unipolar, NRZ, RZ, Manchester and Differential Manchester Transmission Modes: Parallel Transmission, Serial Transmission –	<b>8</b>

	Asynchronous and Synchronous Multiplexing: FDM, TDM, WDM Switching: Circuit Switching, Message Switching, Packet Switching	
<b>4</b>	<b>The Data Link Layer</b> Framing: Character Count, Byte Stuffing, Bit Stuffing, Physical Layer Coding Violations Error Control: Hamming Code and CRC Elementary data link protocols: Simplex, stop & wait protocol, Simplex protocol for noisy channel Sliding Window Protocols: 1-bit sliding window protocols, Pipelining, Go-Back N, Selective Repeat	<b>8</b>
<b>5</b>	<b>Medium Access Control layer</b> Random Access Protocols: ALOHA – pure and slotted, CSMA: 1-persistent, p-persistent and non-persistent, CSMA/CD, CSMA/CA Controlled Access: Reservation, Polling Token Passing Channelization: FDMA, TDMA, CDMA	<b>8</b>
<b>6</b>	<b>The Network layer</b> Design Issues: Store-and-forward packet switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection Oriented Service, Comparison of Virtual Circuit and Datagram Logical Addressing: IPV4 Addresses- Address Space, Notations, Classful Addressing, Classless Addressing, Network Address Translation (NAT) IPV6 Addresses -Addressing Structure, Address Space IPV4 Protocol: Datagram Format, Fragmentation, Checksum, Options Routing Concepts: Properties of routing algorithm, Comparison of Adaptive and Non-Adaptive Routing Algorithms Congestion Control: General Principles of Congestion Control, Congestion Prevention Policies	<b>10</b>
<b>7</b>	<b>The Transport layer</b> Process-to-Process Delivery: Client Server Paradigm, Multiplexing and Demultiplexing, Connectionless Vs Connection-Oriented Service, Reliable Vs Unreliable User Datagram Protocol UDP: Datagram Format, Checksum, UDP operations, Use of UDP Transmission Control Protocol (TCP): TCP Services, TCP Features, TCP Segment, TCP Connection, Flow Control, Error Control TCP Congestion Control: Slow Start Mechanism, Introduction to SCTP	<b>8</b>
<b>8</b>	<b>The Application Layer</b> Domain Name System (DNS): Name Space, Domain Name Space, Distribution of Name Space, DNS in the Internet, Name – Address Resolution E-MAIL: Architecture, User Agent, Message Transfer Agent-SMTP, Message Access Agent-POP, IMAP, Web Based Mail File Transfer Protocol (FTP): Communication over control connection, Communication over Data Connection, Anonymous FTP	<b>8</b>

	WWW: Architecture, WEB Documents HTTP: HTTP Transaction, Persistent and Non-Persistent Connection, Proxy Server	
--	--	--

### **Learning Resources**

1. Tanenbaum, Computer Networks, ISBN:788177581652, Pearson, 4th Edition, 2007
2. Behrouz Forouzan, Data Communication and Networking, TATA McGraw Hill, Fourth Edition, 2011
3. Halsall / Kulkarni, Computer Networking and the Internet, ISBN:9788177584752, Pearson, 5th Edition, 2007
4. Irvine, Data Communications and Networks: An Engineering Approach, ISBN:9788126507658, Wiley India, 2007
5. Gouda, Elements of Network Protocol Design, ISBN:9788126516476, Wiley India, 2009
6. Lary Peterson, Computer Networks-A Systems Approach, 5e, ISBN :9789380501932, Elsevier, 2012

### **Extra Reading:**

1. The concept of router, types and configuring routers
2. The types of switches, how to configure switches
3. How to create domains
4. Concept of security system and configuring firewall.

## F.Y. M.Sc. Semester II

<b>CSA-552</b>	<b>Algorithms and Data Structure</b>	<b>Credits: 04</b> <b>Hours:60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Understand the concept of Dynamic memory management, data types, algorithms, Big O notation	
CO2	Understand basic data structures such as arrays, linked lists, stacks and queues.	
CO3	Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data	
CO4	Identify the different use case where appropriate data structure could be used.	
CO5	Analyse the various sorting and searching algorithms.	
CO6	Write different program to utilize the different data structure.	

Unit No	Contents	No. of Hours
<b>1</b>	<b>Basic Concepts</b> Data Structure, ADT, Algorithms, Time and Space Complexity, Array – basic operations, concept of multi-dimensional array, Polynomial operations using Array, Sparse Matrix.	<b>10</b>
<b>2</b>	<b>Stack and Queue</b> Stack ADT: basic operations, Queue ADT: basic operations, types of Queue, Applications	<b>11</b>
<b>3</b>	<b>Linked List</b> Singly Linked List: concept, representation and operations, Circular Linked List, Polynomial and Sparse Matrix operations using LL, Doubly Linked List, Applications	<b>13</b>
<b>4</b>	<b>Tree and Graph</b> Basic concepts and terminologies, Binary Search Tree and Heap, Disjoint Set, Graph: concept and terminologies, Concept of BFS, DFS, Spanning Tree, Connected Components.	<b>12</b>
<b>5</b>	<b>Searching and Sorting</b> Sequential Search and Binary Search, Insertion Sort, Heap Sort, Radix Sort, External Sorting: k-way merging approach	<b>14</b>

**Learning Resources:**

1. Thareja Reema, Data Structures Using C, Oxford University Press.
2. Sahni Horwitz,, Freed Anderson, Fundamentals of Data Structures in C, University Press
3. Y Langsam , MJ Augenstein and A.M , Tanenbaum Data Structures using C and C++ , Prentice Hall India,
4. Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. Introduction to Algorithms. 3rd ed. MIT Press
5. Data structures: A Pseudo code Approach with C, 2nd edition, R.F.GilbergAndB.A.Forouzan, Cengage Learning.
6. Data structures and Algorithm Analysis in C, M.A.Weiss, Pearson.

**E- BOOKS**

1. <https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf>

**MOOC**

1. <https://nptel.ac.in/courses/106102064>

<b>F.Y. M.Sc. Semester II</b>		
<b>CSA-553</b>	<b>Java Programming</b>	<b>Credits: 04</b> <b>Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	List object-oriented programming concepts.	
CO2	Illustrate the programming constructs of Java, different classes, file handling, thread models and different Java libraries.	
CO3	Examine the usage of interfaces, packages, and exceptional handling.	
CO4	Analyse standard API library of Java.	
CO5	Determine the standard Java packages to understand database programming, collection and web development using Java.	
CO6	Write complex Java programs based on the concept of Swing and web development.	

<b>Unit No.</b>	<b>Contents</b>	<b>No. of Hours</b>
<b>1</b>	<b>Introduction to Java Language</b> History and Evolution of Java, JDK Environment, Java Tools, Java Byte Code OOP Principles, Java Programming – Keywords, Data Types, Variables, Operators, Naming Conventions, Type Casting, Control Statements Arrays	<b>3</b>
<b>2</b>	<b>Object Oriented Concepts of Java</b> Introducing classes and objects Constructors (All types), Garbage Collection and finalize () method Inheritance Basics, Types of Inheritance, Implementation of polymorphism: Method Overloading and Method Overriding Nested and Inner classes, Modifiers and Access Control Specifiers Final variables, methods and classes Abstract methods and classes, Interfaces, Creating and Importing Packages	<b>8</b>
<b>3</b>	<b>Exception Handling</b> Dealing Errors, Exception class, Checked and Unchecked, exception, Catching exception and exception handling, Creating user defined exceptions	<b>5</b>



<b>4</b>	<b>I/O and Files</b> I/O Basics, Exploring java.io package Streams – Byte Streams and Character Streams and their classes, The Predefined Streams, Reading Console Input, Writing Console output, The PrintWriter Class Primitive Type Wrappers: Number, Double and Float, Byte, Short, Integer and Long, Character, Boolean, Void ,Reading and Writing Files, Automatically Closing a File, The I/O Classes and Interfaces, File handling in Java, Strings in java : String Constructors, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data conversion using valueOf (), StringBuffer class	<b>9</b>
<b>5</b>	<b>Multithreaded Programming</b> The Java Thread Model, The Main Thread,Creating a Thread, Creating Multiple Threads, Inter-thread Communication Suspending, Resuming and Stopping Threads	<b>5</b>
<b>6</b>	<b>GUI in Java</b> Event Handling: Event Handling Mechanism, Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter Classes, Inner classes, Anonymous Inner Classes ,Layout Managers Windows Fundamentals, working with Frame Windows, Working with Fonts Swing Features ,Model View Controller Architecture for Swing, Components & Containers ,Swing Components	<b>8</b>
<b>7</b>	<b>Database Programming</b> The design of JDBC JDBC configuration Types of drivers Executing SQL statements ,Query execution, Batch execution Scrollable and updatable result sets Rowset, Metadata, transactions ,Swing application using database	<b>7</b>
<b>8</b>	<b>Collections</b> Introduction to the Collection framework (Interfaces, Implementation, and algorithms) Interfaces Collection classes: Set, List, Queue and Map Set: HashSet, TreeSet, and LinkedHashSet Interfaces such as Lists, Set, Stack, LinkedList, Comparator, Iterator, Hash table	<b>5</b>
<b>9</b>	<b>Web development using JSP</b> Introduction to JSP JSP Architecture JSP Directives ,JSP scripting elements Default objects in JSP JSP Actions ,JSP with Database Error handling in JSP Session tracking techniques in JSP ,JSON object	<b>10</b>

### Learning Resources:

1. Cay S. Horstmann, Gary Cornell, Core Java Volume-II-Advanced Features, Eighth Edition, Prentice Hall, Sun Microsystems Press, 2007.
2. Ivan Bayross, Commercial web development using java 2.0, BPB, 2007.
3. Steven Horlzner , Java 2 programming black books, 2006.
4. Herbert Schildt(5th edition), Complete reference Java, 2002.
5. Jason Hunter,O'Reilly, Java servlet Programming, 2001.

<b>F.Y. M.Sc. Semester II</b>		
<b>CSA-554</b>	<b>Dot Net Framework</b>	<b>Credits: 04 Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	State the fundamentals of Dot Net architecture	
CO2	Represent Object-Oriented principles applied in real life problems.	
CO3	Demonstrate the usage of Reflection and Dynamic programming features.	
CO4	Analyse the problem statement and choose appropriate features / technologies to provide solutions and communicate observations.	
CO5	Identify different approaches for building service-oriented applications.	
CO6	Evaluate appropriate data storage and data access strategy based on type, size of the data and need for data security.	

<b>Unit No.</b>	<b>Contents</b>	<b>No. of Hours</b>
<b>1</b>	<b>Introduction to DOT NET FRAMEWORK</b> What is Framework? ,Architecture of Dot Net Framework ,Common Language Runtime ,Common Type System (CTS) ,Common Language Specification (CLS) ,JIT Compilers ,Base Class Library ,Event Driven Programming	<b>12</b>
<b>2</b>	<b>Introduction to C#</b> Language Fundamentals: Data type and Control Constructs, Value and Reference Types, Boxing, Arrays, String class and its various operations, Functions      Object Oriented Concepts: what is OOP? , Defining classes and Objects, Access modifiers, Constructors, Inheritance, Interface, Abstract Class, Method Overloading and Overriding, Delegates.	<b>16</b>
<b>3</b>	<b>Introduction to ASP.NET</b> What is ASP.NET? ,ASP.NET Page Life Cycle , Architecture of ASP.NET ,Forms, WebPages, HTML forms, Request & Response in Non-ASP.NET pages ,Using ASP.NET Server Controls ,Overview of Control structures ,Functions ,HTML events: ASP.NET Web control events, Event driven programming ,Introduction to Web forms: Web Controls, Server Controls, Client Controls, Navigation Controls, Validations, Master Page, State Management Techniques	<b>18</b>

<b>4</b>	<p><b>Architecture of Ado.Net</b></p> <p>Basics of Ado.net: Connection Object, Command Object, Dataset, Data Table, Data Reader Object, Data Adapter Object ,Datagridview &amp; Data Binding: Insert, Update, Delete records. ,Navigation Using Data Source</p>	<b>14</b>
----------	---	-----------

**Learning Resources:**

1. Beginning Visual C#, WroxPublication
2. BeginningASP.NET3.5,WroxPublication
3. ProgrammingASP.NET3.5byJesseLiberty,DanMaharry,DanHurwitz,O'Reilly
4. Programming Microsoft Visual Basic.NET – Francesco Balena
5. The Complete Reference -Visual Basic .NET – Jeffrey R. Shapiro
6. ADO.NET Examples and Best Practices for C# Programmers, By Peter D, Blackburn, William

<b>CSA-560</b>	<b>The Art of Software Crafting - (Project)</b>	<b>Credits: 02</b>
<b>Course Outcomes (COs)</b>		
<b>On completion of the course, the students will be able to:</b>		
<b>CO1</b>	List and define different problem statements.	
<b>CO2</b>	Explain different software development life cycle models and approaches to solve the problem of a project.	
<b>CO3</b>	Apply resource management skills for the project.	
<b>CO4</b>	Integrate the modules using different techniques and tools.	
<b>CO5</b>	Decide and draw relevant diagrams related to the concerned problem.	
<b>CO6</b>	Prepare the solutions through presentations and technical reports for the concerned problem.	

<b>Sr. No.</b>	<b>Details</b>
<b>I</b>	Selection of Problem Statement
<b>II</b>	Collection of Synopsis
<b>III</b>	Design the problem solution
<b>IV</b>	Implementation of design and refinement if needed
<b>V</b>	Working Progress Report – I
<b>VI</b>	Working Progress Report – II
<b>VII</b>	Working Progress Report – III
<b>VIII</b>	Final report writing and presentation

1. Object Oriented System Development - Ali Bahrami McGRAW-HILL International Edition, 2017.
2. UML in Nutshell, O'reilly Publication, 2015.
3. Software Engineering by Roger Pressman (6th edition), 2009.
4. The Unified Modeling Language user guide by Grady Booch, James Rumbaugh, Ivar Jacobson, 2005.
5. Object Oriented Modeling and Design with UML by James Rumbaugh, Michael Blaha, 2004.
6. UML 2 Bible by Tom Pender, 2002.
7. Object-Oriented Software Engineering: A Use Case Driven Approach by Ivar Jacobson, 1992.

<b>F.Y. M.Sc. Semester II</b>		
<b>CSA- 570</b>	<b>Practical III (Based on Java Programming &amp; Data Structures using C)</b>	<b>Credits: 02 Hours: 60</b>
<b>Course Outcomes (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Identify different packages and interfaces of Java to implement concepts of file streams and object-oriented programming.	
CO2	Illustrate test cases and validations required while developing Java based applications.	
CO3	Apply JDBC to provide a program level interface for communicating with databases using Java programming.	
CO4	Explain Java collection API and Java standard class library.	
CO5	Review different multithreading programming aspects.	
CO6	Write swing-based GUI applications and different web based JSP pages for computer applications as per user demand.	

<b>Sr. No.</b>	<b>Title of the Experiment or Practical</b>
<b>1</b>	Classes, Constructor, and static members
<b>2</b>	Packages, Interface
<b>3</b>	Inheritance and Abstract class
<b>4</b>	Multithreading, File handling
<b>5</b>	Event Driven applications using Swing components
<b>6</b>	JDBC
<b>7</b>	JSP
<b>8</b>	Array, Stack and Queue using C
<b>9</b>	Link List using C
<b>10</b>	Tree using C
<b>11</b>	Graph using C
<b>12</b>	Searching and Sorting using C

### **Learning Resources:**

1. Cay S. Horstmann, Gary Cornell, Core Java Volume-II-Advanced Features, Eighth Edition, Prentice Hall, Sun Microsystems Press, 2007.
2. Ivan Bayross, Commercial web development using java 2.0, BPB, 2007.
3. Steven Horlzner , Java 2 programming black books, 2006.
4. Herbert Schildt(5th edition), Complete reference Java, 2002.
5. Jason Hunter,O'Reilly, Java servlet Programming, 2001.

<b>F.Y. M.Sc. Semester II</b>		
<b>CSA-570</b>	<b>Practical III ( Based on Dot NET Framework &amp; &amp; Data Structures using C)</b>	<b>Credits: 02 Hours: 60</b>
<b>Course Outcome (COs)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	State fundamentals of Dot Net architecture, C# Programming language and User Interface (UI) designing.	
CO2	Illustrate various Object-Oriented principles applied in real life problems. Demonstrate the use of CSS to design a rich UI.	
CO3	Design, develop professional console based and form-based .NET applications. Design and implement various JavaScript methods to query a web page.	
CO4	State the fundamentals of ASP.NET Design the web interface for applications	
CO5	Design and develop window based and web-based .NET applications.	
CO6	Design and Implement database connectivity using ADO.NET for C# and ASP.	

Sr. No.	Title of Experiment/ Practical
1	Working with String and String Builder ,array
2	Delegates: Single Cast and multicast , Exception Handling, Generic collections
3	Forms and web Page in ASP.NET
4	Control Structures and functions in ASP.NET
5	HTML Event Handling & Basics of Ado.NET
6	Datagridview Data Binding
7	Navigation Using Data Source
8	Array, Stack and Queue using C
9	Link List using C
10	Tree using C
11	Graph using C
12	Searching and Sorting using C

### **Learning Resources:**

1. Mark J. Price, C# 8.0 and .NET Core 3.0, Packt Publishing
2. Andrew Stellman, Head First C#: A Learner's Guide to Real-World Programming with C#, XAML, and .NET, O'Reilly
3. Murach, Joel, Murach's ASP.NET Core MVC, Mike Murach & Associates
4. Bill Hamilton, ADO.NET 3.5 Cookbook: Building Data-Centric .NET Applications, O'Reilly

<b>F.Y. M.Sc. Semester II</b>		
<b>CSA-571</b>	<b>Web Development</b>	<b>Credits: 02</b> <b>Hours: 60</b>
<b>Course Outcome (CO)</b> <b>On completion of the course, the students will be able to:</b>		
CO1	Define basic terms and concepts related to web development	
CO2	Develop web pages using the HTML and CSS features with different layouts as per need of applications.	
CO3	Develop web applications using HTML, CSS and JavaScript	
CO4	Use the JavaScript to develop the dynamic web pages.	
CO5	Analyse the responsiveness of a web page using jQuery.	
CO6	Create, design and develop attractive user Interfaces for web applications.	

<b>Unit No.</b>	<b>Contents</b>
<b>1</b>	<b>HTML</b> Fundamentals/ Basic HTML, Text formatting on Web Pages, Images, creating hyperlinks, complex image maps, tables and nested tables, inserting web page, Setting & modifying field properties, Validating HTML
<b>2</b>	<b>CSS</b> Introduction, Features and benefits of CSS, CSS Syntax, External Style Sheet using <link>, Multiple Style Sheets, Selectors: ID Selectors, Class Selectors, Grouping Selectors, Universal Selector, Descendant I Child Selectors, Attribute Selectors, CSS -Pseudo Classes. Text Fonts, text-decoration, text-transform and animations Box Model: Borders & Outline, Margin & Padding, Height and width, CSS Dimensions Embedding Media
<b>3</b>	<b>JavaScript</b> Introduction to JavaScript, JavaScript Engines, Values, Variables and Operators, Variable Mutation, Basic Operators, Operator Precedence, JavaScript Types, Types Definition, Types in JavaScript, Objects, Type Conversion and Coercion, Static vs Dynamic Type Checking.  JavaScript Conditionals: Introduction to Conditionals, Conditionals in JavaScript, Ternary Operators and Conditionals.

	<p>JavaScript Arrays: Introduction to Arrays, Declaring and Mutating Arrays, Array Methods and Properties, Replication with Array Methods, Multi -dimensional Arrays.</p> <p>JavaScript Functions: Introduction to Functions, Functions in JavaScript, Nested Functions in JavaScript, Function as an Argument, Function as the Returned Object, Method of Adding Interactivity to a Web Page, Creating Dynamic Web Pages; Concept of Java Scripting the Forms</p>
<b>4</b>	<p><b>jQuery</b> Introduction, Installing &amp; Configuration, jQuery Syntax, Selectors, Events, jQuery Callback</p>
<b>5</b>	<p><b>Databases and Web Storage</b> NoSQL, Database Connectivity, In-memory data stores, Web storage</p>

#### **Learning Resources:**

1. Chris Bates Web Programming: Building Internet Applications Wiley
2. HTML 5 Black Book Dreamtech Press
3. Beginning JavaScript By Jeremy McPeak and Paul Wilton(Wrox), 2009
4. Head First HTML5 Programming, Building Web Apps with JavaScript By Eric Freeman, Elisabeth Robson (O'Reilly), 2011

#### **Web References:**

1. [https://www.w3schools.com/whatis/whatis\\_fullstack\\_js.asp](https://www.w3schools.com/whatis/whatis_fullstack_js.asp)
2. [https://www.tutorialspoint.com/the\\_full\\_stack\\_web.../index.asp](https://www.tutorialspoint.com/the_full_stack_web.../index.asp)
3. [https://www.w3schools.com/html/html5\\_intro.asp](https://www.w3schools.com/html/html5_intro.asp)