Assam University, Silchar

Four Year Undergraduate Programme

Implemented under NEP 2020

Effective from the Academic Year 2023-24

Syllabus of Computer Application
(4th Year)

Approved in the 99th meeting of the Academic Council on 22.05.2025 vide Resolution No AC:99:05-25:5

Programme Specific Outcome

Bachelor in Computer Application with Honours/Honours and Research

Computer Application (CA) has been evolving as an important branch of Science and Technology in last two decade and it has carved out a space for itself like Computer Science and Engineering. Computer Application spans theory and more application and it requires thinking both in abstract terms and in concrete terms. The ever-evolving discipline of Computer Application has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers and its applications, but finding a solution requires both computer science expertise and knowledge of the particular application domain. BCA and BCA (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career. BCA and BCA (Hons) aims at laying a strong foundation of computer application at an early stage of the career. There are several employment opportunities and after successful completion of BCA, graduating students can fetch employment directly in companies as programmer, Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The present Curriculum Framework for BCA degrees is intended to facilitate the students to achieve the following.

1. Students will have a comprehensive understanding of computer science principles, programming languages, software development methodologies, and data structures.
2. Students will be proficient in one or more programming languages and possess the ability to develop software applications, write efficient code, and solve programming problems.
3. BCA graduates will have strong analytical and problem-solving skills, allowing them to identify and resolve complex computing problems through the application of logical reasoning and critical thinking.
4. Graduates will be capable of designing, developing, and testing software applications using appropriate software engineering principles and methodologies.
5. Students will acquire knowledge and skills in designing and managing databases, including the ability to create database schemas, query data, and ensure data integrity.
6. BCA graduates will have effective oral and written communication skills, enabling them to collaborate with clients, understand requirements, and document software projects accurately.
7. Students will be adept at working in multidisciplinary teams, demonstrating the ability to communicate, cooperate, and contribute to team projects effectively.
8. Graduates will understand the principles of information security and possess knowledge of techniques to secure computer systems, networks, and applications.
9. BCA graduates will be aware of ethical considerations related to computer science and information technology and demonstrate a commitment to professional and ethical practices.
10. Students will be equipped with a strong foundation that allows them to adapt to emerging technologies, learn new programming languages, and continuously update their skills to keep pace with the evolving field of computer science.

Table 1: Semester wise list of Computer Application DSC Courses

Semester	Course Code	Title of the Courses	Credits
I	CADSC101	Fundamentals of Information Technology	3
	CADSC102	Discrete Mathematics	3
II	CADSC151	Data Structure	3
	CADSC152	Lab on Data Structure	3
III	CADSC201	Computer Organization and Architecture	4
	CADSC202	Operating System	4
IV	CADSC251	Programming with Java	4
	CADSC252	Database Management System	4
	CADSC253	Lab on Java Programming & DBMS	4
V	CADSC301	Computer Graphics and C++	4
	CADSC302	System Analysis and Design	4
	CADSC303	Lab on Computer Graphics and C++	4
VI	CADSC351	Computer Network and Internet Technology	4
	CADSC352	E-Commerce and Cyber Sceurity	4
	CADSC353	Programming with PHP	4
	CADSC354	Lab on PHP & Network Programming	4
VII	CADSC401	Design and Analysis of Computer Algorithms	4
	CADSC402	Theory of Computation and Compiler Design	4
	CADSC403	Artificial Intelligence	4
	CADSC404	Lab on DACA & Compiler Design	4
VIII	CADSC451	(A) Research Methodology OR (B) Software Engineering	4
	CADSC452	(A) Image Processing OR (B) Data Analytics	4
	CADSC453	Natural Language Processing	4
	CADSC454	(A) Internet of Things OR (B) Cloud Computing	4
	CADSC455	Research Project/Dissertation	12

Table 2: Semester wise list of Computer Application DSM Papers

Semester	DSM1/DSM2	Course Code	Title of Courses	Credits
I	DSM 1	CADSM101	Programming with C	3
II	DSM 2	CADSM151	Programming with C	3
III	DSM 1	CADSM201	Database Management System	4
IV	DSM 1	CADSM251	Lab on C & DBMS	3
	DSM 2	CADSM252	Database Management System	3
V	DSM 1	CADSM301	Computer Graphics	3
	DSM 2	CADSM302	Computer Graphics	3
VI	DSM 2	CADSM351	Lab on C & DBMS	4
VII	DSM 1	CADSM401	Internet Technology	4
VIII	DSM 2	CADSM451	Internet Technology	4

Table 3: Semester wise list of Computer Application SEC Courses

Semester	Course Code	Title of Courses	Credits
I	CASEC101	Programming with C	3
II	CASEC151	Python Programming	3
III	CASEC201	Web Programming	3

Table 4: Semester wise list of Computer Application IDC Courses

Semester	Course Code	Name of the Paper	Credits
I	CAIDC101	Fundamentals of Information Technology	3
II	CAIDC151	Programming Fundamentals with C	3
III	CAIDC201	Introduction to Web Designing & Cyber Security	3

Syllabi of Computer Application DSC Courses

Semester : VII
Course Type : DSC
Course Code : CADSC401
Name of the Course : Design and Analysis of Computer Algorithms
Learning level : Advanced Course
Credits : 4
Contact Hours : 60
Total Marks : 100
End Semester Marks : 70
Internal Marks : 30

Course Objectives: The course objective is to

1. *Learn fundamental techniques for designing efficient algorithms, including divide and conquer, greedy methods, dynamic programming, and backtracking*
2. *Analyzing Algorithm Efficiency using asymptotic notation (Big O, Omega, Theta)*
3. *Study graph algorithms such as BFS, DFS, shortest path and minimum spanning tree (Kruskal, Prim).*
4. *Understand P vs NP problems, reducibility, and heuristic approaches for solving complex problems.*

UNIT-I

Introduction: Writing structured programs, Analyzing Algorithms, Stacks and Queues, Trees, Heaps and Heap sort, Graphs, Hashing.

UNIT-II

Divide and Conquer: The general concept, binary search, finding the maximum and minimum, merge sort, quick sort, selection sort, strassen's matrix manipulation, greedy methods, minimum spanning trees.

UNIT-III

Dynamic Programming: The general method, multi stage graphs, all pairs shortest paths, optimal binary search trees, reliability design, the travelling sales person problem, flow shop scheduling.

UNIT-IV

Basic search and traversal techniques: The techniques, code generation, AND/OR graphs, game trees, biconnected components, depth first search and breadth first search

UNIT-V

Backtracking: the queens problems, introduction to NP- Hard and NP- Complete problems.

***Course Learning Outcomes**: After successful completion of the course, the students will be able to:*

1. *Analyze Algorithm Efficiency i.e. evaluate the time and space complexity of algorithms using asymptotic notations (Big O, Ω, Θ) and differentiate between worst-case, best-case, and average-case complexities.*
2. *Apply Algorithmic Design Techniques*
3. *Develop algorithms using techniques such as divide and conquer, greedy algorithms, dynamic programming, backtracking, and branch & bound.*
4. *Identify the appropriate algorithmic approach for solving different types of problems.*
5. *Understand NP-Completeness and Computational Complexity i.e. distinguishes between P, NP, NP-complete, and NP-hard problems.*

Text Books:

1. S. Sahni, and Horowitz, Rajasekaran, **Computer Algorithm**, 7th Edition, Galgotia Publications, 2010
2. Horowitz and Sahni, **Fundamentals of Computer Algorithm**, 6th Edition, Galgotia Publications, 2012
3. Aho, Hopcroft and Ullman, **Design and Analysis of Computer Algorithm**, 3rd Edition, Addison-Wesley, 2014.
4. S. E Goodman and S. T Hedet, **Introduction to the Design and Analysis of Algorithm**, 4th Edition, 2009.

Reference Books:

1. S.K Basu, **Design Methods and Analysis of Algorithms**, 2nd Edition, PHI, 2005.
2. Gilles Brassard and Paul Brantley, **Algorithmic theory and Practices**, 3rd Edition, PHI, 2007.
3. Knuth, Fundamental Algorithms, **The Art of Computer Programming**, Vol-1, 3rd Edition, Addison-Wesley, 1997.
4. S. Sahni, **Analysis of Algorithms through C++**, 4th Edition, Tata McGraw Hill, 2009.

Semester	**: VII**
Course Type	**: DSC**
Course Code	**: CADSC402**
Name of the Course	**: Theory of Computation and Compiler Design**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Introduce the theoretical foundations of computer science from the perspective of formal languages.*
2. *Formal methods of computation like automata theory, regular expressions and grammars*
3. *Define different types of formal language and their relationships.*
4. *Introduce the concept of Push Down Automata and the Turing Machine*
5. *Understand to compiler basics, including lexical analysis and syntax trees.*
6. *Understand various phases in the design of the compiler*

Unit-I

Languages and Finite Automata: Alphabets, string, language, Finite Automata, Transitions and Its properties, Acceptability by Finite Automaton, Introduction to Nondeterministic Finite Automata, equivalence of NFA (Nondeterministic Finite Automata) and DFA (Deterministic Finite Automata.

Unit-II

Regular Sets and Regular Grammar: Formal definition, Kleene closure, Algebra of regular expression, Regular languages, closure properties of regular set, Finite automata and Regular Expressions; Context Free Languages: Context free grammars, parse trees, ambiguities in grammars and languages.

Unit-III

Pushdown automata: Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma; Turing Machines: Formal definition of Turing Machine, Transition diagram, Basic structure and working of Turing Machine.

Unit-IV

Overview of compiling process, some typical compiler structures, Symbol Table: The contents of a symbol table, Data structures for symbol tables (ST), design of an ST, The Phases of a Compiler; Lexical Analysis: Role of Lexical Analyzer.

Unit-V

Syntax Analysis: Need and role of the Parser. Intermediate Code Generation: Intermediate Languages, Intermediate Representation Techniques, Statements in three-address code, Implementation of three-address instruction, Three-address code generation. **Code Optimization:** principal sources of optimization.

Course Learning Outcomes: After successful completion of the course, the students will be able to:

1. *Understand the basic properties of formal languages and grammars.*
2. *Differentiate regular, context-free and context-sensitive languages.*
3. *Understand and implement grammars to produce strings from a specific language.*
4. *Understand concepts relating to Push Down Automata and Turing machines*
5. *Understanding lexical analysis and syntax trees is essential for understanding compiler basics.*
6. *Getting a better understanding of intermediate code generation.*
7. *Understanding the generation and optimization of code.*

Text Books:

1. S. P. Eugene Xavier: "**Theory of Automata, Formal Languages and Computation**", New age International Publishers, 2005
2. Daniel I. A. Cohen: "**Introduction to Computer Theory**", John Wiley, 1996
3. Hoperoft, Aho, Ullman: "**Introduction to Automata theory, Language & Computation**", 3rd Edition, Pearson Education, 2006.
4. Alfred V Aho, R.Sethi, D. Ullman, **Compilers - Principles, Techniques and tools**, Pearson, 2nd Edition, 2014
5. Santanu Chattopadhyaya, "**Compiler Design**", PHI, 2nd Edition, 2022.

Reference Books:

1. Lewis & Papadimitriou: "Elements of the Theory of Computation", PHI, 1997.
2. P. Linz: "An Introduction to Formal Language and Automata", 5th edition, Jones and Bartlett Publishers, Inc., USA, 2012.
3. K. L. P. Misha, "Theory of Computer Science: Automata, Languages and Computation" PHI, 3rd Edition, 2006
4. Y. N. Srikant, & P. Shankar, "The Compiler Design Handbook: Optimizations and Machine Code Generation", CRC Press, 2nd Edition, 2018.
5. K. Muneeswaran, "Compiler Design", Oxford University Press, 2012

Semester	**: VII**
Course Type	**: DSC**
Course Code	**: CADSC403**
Name of the Course	**: Artificial Intelligence**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Introduce the historical evolution of Artificial Intelligence (AI), its foundations and applications.*
2. *Explain different search techniques.*
3. *Introduce the concepts of propositional and predicate logic and their applications in AI.*
4. *Explore the scope, potential, limitations, and implications of intelligent systems*

Unit-I

Introduction: Introduction to Artificial Intelligence, History of AI, Risks and Benefits of AI, Brief discussion of applications of AI (Expert System, Natural Language Processing, Speech and Pattern Recognition etc.), Importance of AI. Introduction to Intelligent Agents, their structure, behavior and environment.

Unit-II

Problem Solving and Searching Techniques, Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem.

Unit-III

Knowledge Representation: Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Various approaches used in knowledge representation, Issues in knowledge representation.

Unit-IV

Machine Learning: Forms of learning - Supervised Learning, Unsupervised Learning, Reinforcement Learning; Learning Probabilistic Models, Deep Learning.

Unit-V

Application areas and AI languages: Applications of AI in Engineering, AI languages and their important characteristics, problem solving using Programming in Logic (PROLOG).

Course Learning Outcomes: After successful completion of the course, the students will be able to:

1. *Demonstrate fundamental understanding of the history of Artificial Intelligence (AI), its foundations and applications*
2. *Apply basic principles of AI techniques in solutions that require problem solving, inference, perception, knowledge representation, and learning.*
3. *Use propositional and predicate logic to solve problems in AI.*
4. *Demonstrate proficiency developing applications in an AI programming language*

Text Books:

1. David W. Rolston, "**Principles of Artificial Intelligence and Expert System Development**", McGraw Hill, 2019.
2. Elaine Rich, Kevin Knight: "**Artificial Intelligence**", Tata McGraw Hill, 2019.
3. D.W. Patterson, "**Introduction to AI and Expert Systems**", PHI, 2018.
4. Russell, S., Norvig, P., "**Artificial Intelligence: A Modern Approach**". Create Space Independent Publishing Platform, 2016.

Reference Books:

1. W.F. Clocksin and Mellish, "**Programming in PROLOG**", Narosa Publishing edition, 1991. House, 3rd Edition, 2001.
2. Ivan Bratko, "**Prolog Programming for Artificial Intelligence**", Addison-Wesley, Pearson Education, 3rd Edition, 2000.

Semester	**: VII**
Course Type	**: DSC**
Course Code	**: CADSC404**
Name of the Course	**: Lab on DACA & Compiler Design**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

The objective of laboratory paper on Design and Analysis of Computer Algorithms and Compiler Design is to learn implementation of different algorithms to understand the running time complexity practically. Students will execute different algorithms of incremental methods, divide and conquer methods, dynamic programming, backtracking, heuristic approach and evaluate the

effectiveness. Students will also get the opportunity about designing various components of compiler.

This paper provides practical knowledge of DACA and Compiler Design. List of laboratory programming assignments (not limited to these):

Lab on DACA

Problems related to Design and Analysis of Computer Algorithms should be solved by using the Programming languages C/C++/JAVA (preferably on Unix/Linux/Solaris operating systems environment on a network).

1. Stack and queues, tree, heap and heap sort, graphs and hashing.
2. Divide and conquer method: binary search, merge sort, quick sort, matrix multiplication, minimum spanning tree.
3. Dynamic programming: multistage graphs, all pair shortest paths, optimal binary search trees, traveling salesperson problem, flow shop scheduling.
4. Search and traversal techniques: AND/OR graphs, game trees, bi connected components and depth search.
5. Backtracking: Hamilton cycles, the fast Fourier transform, NP-HARD and NP complete problems.

Lab on Compiler Design

Problems related to Compiler Design should be solved by using the Programming languages C/C++/JAVA as well as various tools for Compiler Construction and Design like LEX, YACC etc.

1. Construction of a lexical analyzer and LL(1) parser for a subset of FORTRAN/PASCAL/C/C++ (to be done without using any generator).
2. Construction of a lexical analyzer and LALR(1)/LR(1) parser for a subset of C/C++ (generators like LEX, YACC to be used).
3. A construction of a translator from a high level to an intermediate language which is also a very simple subset of C (The correctness of this translation may be checked by compiling this intermediate program by a standard compiler).
4. Construction of a target code generator from the above intermediate language program to the assembly language of a suitable target machine (e. g. Intel 8088). Addition of rudimentary code optimization (like peep-hole)/jump optimization.
5. Register optimization to the generated compiler.
6. Experiments with incorporation of debugging features.

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

1. *Implement different data structures and analyze the running time.*
2. *Understand the effectiveness of divide and conquer methods by implementing different standard algorithms.*

3. *Compare the performance of Dynamic programming with the divide and conquer method.*
4. *Design different algorithms for NP-HARD and NP complete problems.*
5. *Design various components of compiler*

Semester	**: VII**
Course Type	**: DSM**
Course Code	**: CADSM401**
Name of the Course	**: Internet Technology**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Understand basic Internet architecture, protocols, and standards*
2. *Explore networking concepts like TCP/IP, HTTP, FTP, and DNS*
3. *Learn about web development basics: HTML, CSS, JavaScript and PHP*

UNIT I

Internet, structure of internet, history of internet, Internet protocol: TCP/IP, SLIP, PPP, Network and network devices, Addressing in Internet - DNS, domain name and their organisation, understanding the Internet protocol address, Client-server concept- architecture and application. Evolution of www, basic features, servers, http, FTP, URL, search engine, searching categories, hypertext.

UNIT II

HTML: Basic HTML, HTML tags, creating list in HTML, hyperlinks, multimedia, HTML forms, tables in HTML, frames in HTML, image maps, style sheets in HTML. DHTML, XML-Introduction, Need for XML, Advantages, simple XML programs, DTD.

UNIT III

Creating interactive and dynamic web pages with JavaScript: Client-side scripting languages, JavaScript overview; constants, variables, operators, expressions and statements; user-defined & built-in-functions; properties and methods of built-in objects, client-side form validation.

UNIT IV

Introduction to PHP: Server side scripting language, Evaluation of PHP, Basic Syntax, Defining variable and constant, PHP Data type, Operator and Expression. Decisions and loop Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with HTML

UNIT V

Handling Html Form with PHP: Capturing Form, Data Dealing with Multi-value filed, and generating File uploaded form, redirecting a form after submission. Database Connectivity with MySQL: Introduction to RDBMS, Connection with MySQL Database, Performing basic database operation.

Course Learning Outcomes: After successful completion of the course, the students will be able to:

1. *Explain the history, evolution, and architecture of the Internet*
2. *Develop basic web pages using HTML, CSS, and JavaScript.*
3. *Understand how web servers, browsers, and databases interact.*

Text Books:

1. M.L Young , **The Internet: Complete**; Tata McGraw Hill, 3rd Edition, 2006.
2. J. Jaworski, **Mastering JavaScript and Jscript**, BPB Publication, 2nd Edition, 2011.
3. D. Godman, **Dynamic HTML: The definitive references**, Shroff Publishers, 2nd Edition, 2009.
4. Steven Holzner, **PHP: The Complete Reference Paperback**, McGraw Hill Education (India), 2007.

Reference Books:

1. Daniel Minoli, **Internet and Internet Engineering**, Tata McGraw-Hill Edition, 2nd Edition, 2011.
2. Luke Welling, Laura Thompson, **PHP and MySQL Web Development**, 4th Edition, Addition Paperback, Addison-Wesley Professional, 2008.

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC451 (A)**
Name of the Course	**: Research Methodology**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

(*To be prepared at the university level)**

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC451 (B)**
Name of the Course	**: Software Engineering**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Understand the concept of software engineering*
2. *Introduce the various stages of Software Development Life Cycle*
3. *Introduce the various process models.*
4. *Introduce the software quality, reliability and configuration management*
5. *Learn the design & testing principles to software project development*

Unit-I

Introduction: Importance of software, Software Characteristics, importance of Software engineering , Software Process Framework. Life Cycle Models: Waterfall model, Iterative models, RAD Model, Prototyping Model, Spiral Model.

Unit-II

Requirement Analysis: Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS. **Software Project Management:** Estimation in Project Planning Process, Project Scheduling.

Unit-III

Risk Management: Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan. **Quality Management:** Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and Projects.

Unit-IV

Software Design: Conceptual and Technical Designs, Objectives of software Design, Modularity**:** Module Coupling, Module Cohesion, Relation between cohesion and coupling. Design Strategy**:** Bottom- up, Top-Down. Function oriented design**:** design notations, Functional Procedural layers, DFD, Flowchart, Structure charts, Transform and transaction analysis. Object Oriented Design**:** Basic mechanism, concepts, advantages of OOD, unified modeling language (UML)

Unit-V

Testing Strategies & Tactics: Software Testing Fundamentals, Strategic Approach to Software Testing, Unit testing, Integration testing, Validation Testing, System testing, Black-Box Testing, White-Box Testing and their type, Basis Path Testing.

***Course Learning Outcomes**: After successful completion of the course, the students will be able to:*

1. *Understand the various stages of SDLC, the various process models, software quality, and reliability and configuration management.*
2. *Explain SRS documents, planning, analysis, design and development of software*
3. *Implement, test and validate a system designs and develop the software*

Text Books:

1. Roger S. Pressman: **Software Engineering, A Practitioner's Approach**, 7th Edition, Tata McGraw Hill, 2014.
2. Rajib Mall: **Fundamentals of Software Engineering**, 5th Edition, PHI, 2011.

Reference Books:

1. Pankaj Jalote, **An Integrated Approach to Software Engineering**, Narosa Publishing House, 3rd Edition, 2011.
2. K. K. Aggarwal, Yogesh Singh**: Software engineering**, 3rd Edition, New Age International publishers, 2009.
3. F. Tsui, O. Karam, B. Bernal**: Essentials of Software Engineering**, 4th Edition, Jones & Bartlett, 2006.

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC452 (A)**
Name of the Course	**: Image Processing**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Introduce the concepts of Digital Image Processing and fundamental techniques*
2. *Explain the concepts of Digital Image Transforms*
3. *Understand image compression techniques*
4. *Explain Image segmentation techniques*

Unit-I

Digital image fundamentals - Digital Image representation, Image acquisition, Fundamental steps in Image processing, Elements of digital Image processing systems, Types of Images, Gray level to binary image conversion, Sampling and quantization

Unit-II

Image Transforms: Need for Image transform, Fourier Transform, 2D Discrete Fourier Transform, Properties of 2D FFT, Walsh transform, Hadamard Transform, Discrete cosine Transform, Haar transform, Slant transform, Hotelling transform

Unit-III

Image enhancement: Point processing. Histogram processing. Spatial filtering. Enhancement in frequency domain, Image smoothing, Image sharpening.

Unit-IV

Image Compression: Image Compression Models, Source encoder and decoder, Error free compression, Lossy compression, Image Compression Measures, Huffman Coding.

Unit-V

Image segmentation: Detection of discontinuities, Edge linking and boundary detection, Thresholding, Region oriented segmentation.

Course Learning Outcomes*: After successful completion of the course, the students will be able to:*

1. *Learn image conversion and image transformation techniques*
2. *Apply image enhancement techniques*
3. *Learn image compression models.*
4. *Apply various image processing techniques for real time applications*

Text Books:

1. R. C. Gonzalez, R. E. Woods, "**Digital Image Processing",** 4th Edition, Pearson Education, 2018.
2. A. K. Jain, **"Fundamentals of Digital Image Processing"**, Prentice Hall of India. 1989.

Reference Books:

1. K. R. Castleman, "**Digital Image Processing**", Pearson Education, 1996.
2. Rafael C. Gonzalez, Richard E.Woods, Steve Eddins, "**Digital Image Processing Using MATLAB**", Pearson Education, Inc., 2004.
3. William K. Pratt, **Digital Image Processing**, John Wilely, 3rd Edition, 2004.

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC452 (B)**
Name of the Course	**: Data Analytics**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Explain the basics of data science and its applications*
2. *Learn the strategies of data collection and pre-processing*
3. *Explain data analytics using Python*
4. *Learn the application of AI in data analytics*

Unit-I

Introduction: Introduction to Data Analytics – Evolution of Data Analytics – Data Analytics Roles – Stages in a Data Analytics Project – Applications of Data Analytics in various fields – Data Security Issues. Data Collection Strategies – Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization.

Unit-II

Deep Dive with Excel: Data Connectors in Excel, Cleaning in Power Query Editor, Adding Conditional Columns using Power Query, Editor, Data Modelling and its Importance, Cardinality and Filter Direction in Power Pivot, Pivot Tables in Excel, Charts in Excel, Creating a Dashboard in Excel

Unit-III

Python for Data Analytics: Introduction to Python for Data Analytics, In-built Modules, Pickle Library, Introduction to Numpy, Statistical Functions in Arrays, WebScraping, BeautifulSoup and Requests Library in Python, Extracting Data from Tables, Extracting Data from Multi-Page Websites, Text Analysis using Python.

Unit-IV

Machine Learning: Introduction, Applications of Machine Learning, Linear Regression, Decision Trees, Random Forests and Ensemble Methods, Clustering Algorithms, Dimensionality Reduction.

Unit-V

AI for Data Analytics: Generating code & debugging errors, Exchanging Data Cleaning Skills, Crafting SQL queries, Optimizing Model Selection with AI-driven Methods, Utilizing AI for Anomaly Detection in Data Analysis

Course Learning Outcomes*: After successful completion of the course, the students will be able to:*

1. *Understand the concept of data science and applications*

2. *Understand the methods of data collection and processing*
3. *Perform data analysis using python*
4. *Employ AI for data analytics*

Text Books:

1. Cathy O'Neil and Rachel Schutt , "**Doing Data Science**", O'Reilly, 2015.
2. Fabio Nelli, "**Python Data Analytics**", Apress, Publishing, 1st Edition, 2015

Reference Books:

1. G. Sudhamathy and C. Jothi Venkateswaran, **R Programming: An Approach to Data Analytics**, MJP Publishers, 2019

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC453**
Name of the Course	**: Natural Language Processing**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Learn the basics of human language structure and computational linguistics*
2. *Understand the core concepts and challenges of NLP*
3. *Understand Morphological Analysis part-of-speech tagging and syntactic parsing*
4. *Work with machine translation, speech recognition, and text-to-speech systems.*

Unit I

Introduction to NLP: Some example application, Achievements and brief history, Open problems, Major Goals. Introduction to Language Structure and Language Analyzer: Introduction to Language Structure, Overview of Language Analyzer Requirement of Computational Grammars.

Unit II

Linguistics Resources: Introduction to corpus, elements in balanced corpus, TreeBank, PropBank**. Pre-processor:** Objectives of Processor, Analyzing a text, Discourse Language, Punctuation's, Abbreviations, names, special characters, Need for human Preprocessor.

Unit III

Words and their Analyzer : Introduction, Why Morphological Analysis, Morphological Generation using paradigms, Morphological Analysis using Paradigms, Speedup Morphological Analysis by compilation, Morphological Analyzer – Some Additional Issues.

Unit IV

Part of Speech tagging- Stochastic POS tagging, Hidden Markov Models (HMM), Transformation based tagging (TBL), Handling of unknown words, named entities, multi word expressions. **Rule Base:** Sentences and their analysis, Why Rule Base, Application in Rule Base, Verb Groups, Noun Group, Strategy for Grammar Development, and Semantics in stages.

Unit V

Parsing- Unification, probabilistic parsing, TreeBank. Semantics- Meaning representation, semantic analysis, lexical semantics, WordNet. **Machine Translation:** Introduction, Problems of Machine Translation, Brief History, Possible Approaches, ANGLABHARTI and its importance. Applications of NLP- Spell-checking, Summarization.

Course Learning Outcomes*: After successful completion of the course, the students will be able to:*

1. *Explain the fundamental principles and challenges of NLP.*
2. *Describe the structure of human language and its computational representation*
3. *Perform part-of-speech tagging, named entity recognition (NER), and syntactic parsing*
4. *Build real-world NLP applications such as chatbots, sentiment analysis, text summarization, and machine translation*

Textbook:

1. Daniel Jurafsky and James H Martin. **Speech and Language Processing**, 2nd Edition, Pearson Education, 2009

Reference Books:

1. James A.. **Natural language Understanding**; Pearson Education, 2nd Edition, 1994.
2. Bharati A., Sangal R., Chaitanya V.. **Natural language processing: a Paninian perspective**, PHI, 2000
3. Siddiqui T., Tiwary U. S.. **Natural language processing and Information retrieval**, OUP, 2008

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC454 (A)**
Name of the Course	**: Internet of Things**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Understand the basic concepts of IoT,*
2. *Understand the application areas of IoT.*

3. *Explain the concepts of IoT devices and sensors.*
4. *To understand the future trends of IoT.*

Unit-I

Introduction to IoT: Definition and Concept of IoT, Evolution and Importance of IoT, Components of IoT Ecosystem, IoT Architecture and Communication Protocols, IoT Applications in Different Industries, Ethical and Privacy Considerations in IoT

Unit-II

IoT Devices and Sensors: Types of IoT Devices (Sensors, Actuators, Smart Objects), IoT Device Connectivity (Wired, Wireless, Cellular), IoT Access Technologies, Data Acquisition and Sensor Fusion, Applications protocols for IoT

Unit-III

Data and Analytics for IoT: An introduction to data analytics for IoT, Machine Learning, Big data analytics tools and technology, edge streaming analytics, network analytics.

Unit-IV

IoT Networking and Security Networking Models for IoT (Star, Mesh, etc.), IPv6 and IP Addressing in IoT, Wireless Sensor Networks, Security Challenges in IoT, IoT Security Measures (Authentication, Encryption, etc.), IoT Security Best Practices and Standards

Unit-V

IoT Applications and Future Trends: Smart Cities and Urban IoT, Industrial IoT (IIoT) and Industry 4.0, Healthcare and Wearable IoT, Agriculture and Environmental Monitoring, Home Automation and Consumer IoT, Emerging Trends and Future Directions of IoT.

Course Learning Outcomes*: After successful completion of the course, the students will be able to:*

1. *Explain the Architecture of IoT.*
2. *Understand different types of sensor and IoT network.*
3. *Learn the applications of IoT and its future*

Text Books:

1. D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry, **IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things**, Cisco Press, 2017.
2. Raj Kamal, **Internet of Things (Architecture and Design Principles)**, McGraw Hill Education, 2017.

Reference Books:

1. Arshdeep Bahga and Vijay Madisetti, "**Internet of Things: A Hands-On Approach**", Orient Blackswan Private Limited, 2015

2. Cuno Pfister "**Getting Started with the Internet of Things**" O'reilly Media, INC International Concepts USA, 2011.

Semester : VIII
Course Type : DSC
Course Code : CADSC454 (B)
Name of the Course : Cloud Computing
Learning level : Advanced Course
Credits : 4
Contact Hours : 60
Total Marks : 100
End Semester Marks : 70
Internal Marks : 30

Course Objectives: The course objective is to

1. *Compare the strengths and limitations of cloud computing*
2. *Identify the architecture, infrastructure and delivery models of cloud computing*
3. *Apply suitable virtualization concept.*
4. *Choose the appropriate cloud player, Programming Models and approach.*
5. *Address the core issues of cloud computing such as security, privacy and interoperability*

Unit I

Introduction to Cloud Computing: Evolution and History of Cloud Computing, Introduction to Cloud Computing, Why Cloud Computing is Becoming Highly Important, Features of Cloud Computing, Cloud Computing for various users, Advantages of Cloud Computing, Limitations of Cloud Computing.

Unit II

Cloud Models and Types: The NIST Model, Cloud Cube Model, Deployment Models, Service Models. Layers and Types of Cloud, Components of Cloud Computing, Cloud Computing Service Providers

Software as a Service (SaaS): Software as a Service, Evolution of SaaS, Brief Introductory part of Software as a Service, SaaS Unification Technologies, SaaS Integration Products and Technologies, SaaS Product Selection Criteria, SaaS Integration Services, Advantages of SaaS.

Unit III

Platform as a Service (PaaS): Introduction to PaaS, Evolution of PaaS, PaaS Service Providers- Acquia Cloud, Amazon AWS, Amazon Elastic Beanstalk, Google App Engine, Force.com, PaaS Application Framework, PaaS Operator Verbs, PaaS Developer Verbs, Advantages and Challenges of PaaS.

Unit IV

Infrastructure as a Service (IaaS): Evolution, IaaS Architecture- Advantages and Disadvantages of Infrastructure as a Service, SAN model, IaaS Providers, IaaS Architecture, Advantages and Disadvantages of Infrastructure as a Service.
Data in Cloud : Evolution of Network Storage in Cloud, Data as a Service, Database as a Service, Cloud Based Data Storage, Advantages and Limitations of Cloud Based Storage Solution, Cloud Based Data Storage Service Providers.

Unit V

Virtualization: Introduction to Virtualization and its Technical Evolution, History of Virtualization, Types of Virtual Machines, Advantages of Virtualization, Components of Virtualization, Types of Virtualization.

Course Learning Outcomes*: After successful completion of the course, the students will be able to:*

1. *Explain the core concepts, benefits, and challenges of cloud computing.*
2. *Describe different cloud service models (IaaS, PaaS, SaaS) and deployment models*
3. *Understand cloud architecture, data centers, and virtualization technologies*

Text Books:

1. Dr. Anand Nayyar, **Handbook of Cloud Computing**, 1st Edition, BPB Publication, India, 2019.
2. Anthony T.Velte, Toby J. Velte Robert Elsenpeter, **Cloud computing: A practical approach** - TATA McGraw- Hill, 2010.
3. Michael Miller, **Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online**, Que Publication, 2008

Reference Books:

1. Dan C Marinescu, **Cloud Computing, Theory and Practice**, MK Publisher, Import 4, 2013.
2. Arshadeep Bahga, Vijay Madisetti **Cloud Computing, A Hands on approach**, Vijay Madisetti Publisher, 2014**.**
3. Raj Kumar Buyya, Christenvecctiola, S Tammaraiselvi, **Mastering Cloud Computing, Foundations and Application Programming**, TMH, 2011.

Semester	**: VIII**
Course Type	**: DSC**
Course Code	**: CADSC455**
Name of the Course	**: Research Project/Dissertation**
Learning level	**: Advanced Course**
Credits	**: 12**
Contact Hours	**: 360**
Total Marks	**: 300**
End Semester Marks	**: 210**
Internal Marks	**: 90**

- ✓ ***Honours students not undertaking research will do 3 courses for 12 credits in lieu of a research project / Dissertation***.
- ✓ Students who secure 75% marks and above in the first six semesters and wish to undertake research at the undergraduate level can choose a research stream in the fourth year. They should do a research project or dissertation under the guidance of a faculty member of the University/College. The research project/dissertation will be in the major discipline. The students who secure 160 credits, including 12 credits from a research project/dissertation , are awarded UG Degree (Honours with Research).
- ✓ The research outcomes of their project work may be published in peer-reviewed journals or may be presented in conferences /seminars or may be patented.
- ✓ The final semester will be devoted to seminar presentation, preparation, and submission of project report/dissertation. The project work/dissertation will be on a topic in the disciplinary programme of study or an interdisciplinary topic.
- ✓ Students choose a research component with courses relating to research methodology, advanced courses in theory and applied areas, and seminar presentations. Students may be permitted to carry out a research project or dissertation in another department of the same institution or another institution provided the required facilities are available.

Semester	**: VIII**
Course Type	**: DSM**
Course Code	**: CADSM451**
Name of the Course	**: Internet Technology**
Learning level	**: Advanced Course**
Credits	**: 4**
Contact Hours	**: 60**
Total Marks	**: 100**
End Semester Marks	**: 70**
Internal Marks	**: 30**

Course Objectives: The course objective is to

1. *Understand basic Internet architecture, protocols, and standards*
2. *Explore networking concepts like TCP/IP, HTTP, FTP, and DNS*
3. *Learn about web development basics: HTML, CSS, JavaScript and PHP.*

UNIT I

Internet, structure of internet, history of internet, Internet protocol: TCP/IP, SLIP, PPP, Network and network devices, Addressing in Internet - DNS, domain name and their organisation, understanding the Internet protocol address, Client-server concept- architecture and application. Evolution of www, basic features, servers, http, FTP, URL, search engine, searching categories, hypertext.

UNIT II

HTML: Basic HTML, HTML tags, creating list in HTML, hyperlinks, multimedia, HTML forms, tables in HTML, frames in HTML, image maps, style sheets in HTML. DHTML, XML-Introduction, Need for XML, Advantages, simple XML programs, DTD.

UNIT III

Creating interactive and dynamic web pages with JavaScript: Client-side scripting languages, JavaScript overview; constants, variables, operators, expressions and statements; user-defined & built-in-functions; properties and methods of built-in objects, client-side form validation.

UNIT IV

Introduction to PHP: Server side scripting language, Evaluation of PHP, Basic Syntax, Defining variable and constant, PHP Data type, Operator and Expression. Decisions and loop Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with HTML

UNIT V

Handling Html Form with PHP: Capturing Form, Data Dealing with Multi-value filed, and generating File uploaded form, redirecting a form after submission. Database Connectivity with MySQL: Introduction to RDBMS, Connection with MySQL Database, Performing basic database operation.

Course Learning Outcomes: After successful completion of the course, the students will be able to:

1. *Explain the history, evolution, and architecture of the Internet*
2. *Develop basic web pages using HTML, CSS, and JavaScript.*
3. *Understand how web servers, browsers, and databases interact.*

Text Books:

1. M.L Young , **The Internet: Complete;** Tata McGraw Hill, 3rd Edition, 2006.
2. J. Jaworski, **Mastering JavaScript and Jscript**, BPB Publication, 2nd Edition, 2011.
3. D. Godman, **Dynamic HTML: The definitive references**, Shroff Publishers, 2nd Edition, 2009.
4. Steven Holzner, PHP: **The Complete Reference Paperback**, McGraw Hill Education (India), 2007.

Reference Books:

1. Daniel Minoli**, Internet and Internet Engineering,** Tata McGraw-Hill Edition, 2nd Ed., 2011.
2. Luke Welling, Laura Thompson, **PHP and MySQL Web Development**, 4th Edition, Addition Paperback, Addison-Wesley Professional, 2008.