3-Year BCA
(Bachelor of Computer Application)
Syllabus
[W.E.F. Academic Year 2018-2019]

A three year fulltime semester based UG programme in Bachelor of Computer Application

Total Credit: 116

I	II	III	IV	V	VI	Total
18	21	18	18	23	18	116

Department of Computer Science & Engineering Aliah University
II A/27, New Town

Kolkata – 700156, West Bengal, India.

PO1: Pursue further studies with specialization in Computer Science and Applications and Business Administration.

PO2: Apply knowledge of mathematics, computer science and management in practice.

PO3: Develop skills in computer programming languages, networking, applications and packages, system administration, Web Technologies, and modern IT tools.

PO4: Blend analytical, logical and managerial skills with the technical aspects to resolve real world issues.

PO5: Become employable in various IT companies and government jobs.

PO6: Effectively communicate business issues, management concepts, plans and decisions both in oral and written form.

PO7: Capable of recognizing and solving ethical issues.

.

Semester I:

Introduction to Computers and Application

Code: BCA101
Contracts: 3L
Credits: 3

Course Objectives:

- To develop the programming skills of students
- To know the principles of designing structured programs
- To write basic C programs using
    i) Selection statements
    ii) Repetitive statements
    iii) Functions
    iv) Pointers
    v) Arrays
    vi) Strings

Course Outcomes:

CO 1: Understanding the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming.
CO 2: Write, Compile and Debug programs in C language and use different data types for writing the programs.
CO 3: Design programs connecting decision structures, loops and functions.
CO 4: Explain the difference between call by value and call by address.
CO 5: Understand the dynamic behavior of memory by the use of pointers.
CO 6: Use different data structures and create/ manipulate basic data files and developing Applications for real world problems.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√		√				
CO2		√		√			
CO3		√		√			
CO4		√	√	√			
CO5			√	√			
CO6			√				

Module 1: [12 Hrs]

Background: History of computing, overview of computers, basic organization of the von Neumann machine; instruction fetch, decode, and execution;
Programming languages and the compilation process. Fundamental programming constructs: Syntax and semantics of a higher-level language like C; variables, types, expressions, and assignment; simple I/O; conditional and iterative control structures; functions and parameter passing; structured decomposition.

Module 2: [12 Hrs]

Algorithms and problem-solving: Problem-solving strategies; the concept of an algorithm; properties of algorithms; implementation strategies; concept of recursion; sequential and binary search algorithms; quadratic sorting algorithms (selection, insertion).

Module 3: [12 Hrs]

Fundamental data structures: Primitive types; arrays; records; strings and string processing; pointers and references; runtime storage management. Machine level representation of data: Bits, bytes, and words; binary representation of integers; representation of character data; representation of records and arrays. Introduction to data structures: stacks and queues.

Suggested Books:

1. Parsons and Oja: "Computer Concepts--Illustrated Series", Introductory, 6th Edition, (Concepts Textbook),
2. Beskeen, Cram, Duffy, Friedrichsen, and Reding: "Microsoft Office 2003 --Illustrated Series, Premium Edition, Introductory", (Lab Textbook),

Introduction to Programming

Code: BCA103
Contacts: 3L
Credits: 3

Course Objectives:

- To provide a comprehensive study of the C programming language, stressing upon the strengths of C.
- To provide the students with the means of writing modular, efficient, maintainable, and portable code.

Course Outcomes:

CO 1: Students should be able to write, compile and debug programs in C language.
CO 2: They should be able to use different data types in a computer program.
CO 3: They should be able to design programs involving decision structures, loops and functions.
CO 4: Students should be able to explain the difference between call by value and call by reference.
CO 5: Students should be able to understand the dynamics of memory by the use of pointers.
CO 6: Students should be able to use different data structures and create/update basic data files.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2		√	√			√	
CO3	√	√	√	√	√		√
CO4		√	√	√		√	
CO5		√	√		√		
CO6	√			√	√		√

Module - 1 [12 Hrs]

Introduction: History of Computing, Evolution of Programming Languages, Compilers, Interpreters. **[2L]**

Problem Solving Method: Algorithms and Flowcharts. **[2L]**

Overview of C: Brief History of C, C Standards, Structure of a C Program, C Libraries and Linking, Compiling a C Program. **[2L]**

Expressions: Basic Data Types, Variables, Type Qualifiers, Storage Class Specifiers, Variable Scopes, Constants, Operators, Operator Precedence, Expression Evaluation, Type Conversion in Expressions, Type Casting **[6L]**

Module - 2 [12 Hrs]

Statements: Selection Statements (if, switch-case), Iteration Statements (for loop, while loop, dowhile loop), Jump Statements (return, goto, break, exit, continue). **[4L]**

Arrays and Strings: Single Dimension Arrays, Double Dimension Arrays, Strings, Arrays of Strings, String Library Functions. **[4L]**

Functions: General Form, Function Prototypes, Parameter Passing Mechanisms, Command Line Arguments, Recursion. **[4L]**

Module - 3 [12 Hrs]

Pointers: Pointer Variables, Pointer Operators, Pointer Expressions, Pointers and Arrays, Functions and Pointers, Pointers to Functions, Dynamic Memory Allocation. **[3L]**

Structures, Unions, Enumerations and Typedef: Structures, Arrays of Structures, Structure Pointers, Unions, Bit Fields, Enumerations, Typedef. **[4L]**

Console I/O: Reading and Writing Characters, Reading and Writing Strings, Formatted Console I/O. **[1L]**

File I/O: Data Organization, File Operations, Text Files and Binary Files, Random Access. **[2L]**

The Preprocessor: Preprocessor Directives, Macros, Macro vs. Function, File Inclusion, Conditional Compilation. **[2L]**

Suggested Books:

1. Herbert Schildt: “C: The Complete Reference”, 4th Edition, Tata McGraw Hill, 2000.
2. Stephen Prata: “C Primer Plus”, 5th Edition, SAMS Publishing, 2005.
3. Brian W. Kernighan and Dennis M. Ritchi: “C Programming Language”, 2nd Edition, Pearson Education, 2006.
4. Samuel P. Harbison and Guy L. Steele: “C: A Reference Manual”, 5th Edition, Prentice Hall, 2003.
5. Yashwant Kanetkar: “Let Us C”, BPB Publications, 9th Edition, 2008.
6. K. N. King: “C Programming: A Modern Approach”, 2nd Edition, W. W. Norton and Company, 2008.
7. Andrew Koenig: “C Traps and Pitfalls”, Addison Wesley Professional, 1989.

Digital Logic

Code: BCA105
Contacts: 3L
Credits: 3

Course objectives:

The objectives of this course are to:

1. Introduce the concept of digital and binary systems.
2. To understand the concept of Boolean algebra and various logic gates.
3. Design and analyze combinational logic circuits.
4. Design and analyze sequential logic circuits.
5. To provide knowledge about digital integrated circuits.

Course outcomes:

CO 1: Explain the principles and methodology of digital logic design at the gate and switch level.
CO 2: Define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
CO 3: Understand Boolean algebra and basic properties of Boolean algebra; able to simplify simple Boolean functions by using the basic Boolean properties.
CO 4: Familiar with basic combinational logic circuits: Adder, subtractor, encoder, decoder, comparator etc.
CO 5: Familiar with basic sequential logic components: flip-flops, registers and counters.
CO 6: Understands the concepts of Diodes, transistors, MOS, CMOS etc.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2	√	√	√	√			
CO3	√	√		√			
CO4		√		√			
CO5		√		√			
CO6	√	√	√				

Module - 1: [8 Hrs]

Number Systems, Boolean Algebra & Logic Gates: Binary numbers & Boolean algebra, Venn diagram, Logic gates, Truth Tables and function minimization using algebraic method, Karnaugh map, Quine- Mcclusky method; BCD, ASCII, EBDIC, Gray codes and their conversions, Signed binary number representation with 1"s and 2"s complement methods, Maxterm, Minterm, Representation in SOP and POS forms ; Realization of Boolean functions using NAND/NOR gates, two-level and multilevel logic circuit synthesis.

Module - 2: [10 Hrs]

Combinational circuits: Adder and Subtractor circuits (half & full adder & subtractor); Encoder, Decoder, Comparator, Multiplexer, De-Multiplexer and Parity Generator and checker; ROM, PLA .

Module - 3: [10 Hrs]

Sequential Circuits: Latch, Flip-flop. Design of Flip-flops with logic gates. Counters, Registers. Design and analysis of sequential circuits -Moore and Mealy model description, state diagram and state table – Minimization methods. Memory unit. Racing and Logic hazards. Implementation of hazard free logic circuit. asynchronous sequential circuit synthesis.

Module – 4: [10 Hrs]

Digital Integrated Circuits: Diode as switch. Use of diodes in AND, OR Circuits . Transistor as a switch. RTL, DTL, TTL logic gate circuits. MOS as a switch. Basic MOS inverter. MOS and CMOS logic gates. Fan -in and Fan-out of logic gates, propagation delay, Tristate logic.

Suggested Books:

1. Floyed and Jain: "Digital Fundamentals", Pearson Education.
2. Morries Mano: "Digital Logic Design", PHI.
3. Leach & Malvino: "Digital Principles & Application", 5/e, Tata McGraw Hill.
4. Kharate: "Digital Electronics", Oxford.
5. Bigmell and R.Donovan: "Digital Electronics - Logic & Systems", Cambridge Learning.
6. D.J.Comer: "Digital Logic and State Machine Design", 3/e. OUP.
7. P.Raja: "Digital Electronics", Scitech Publications.
8. R.P.Jain: "Modern Digital Electronics", 2/e , Tata McGraw Hill.
9. H.Taub and D.Shilling: "Digital Integrated Electronics", Tata McGraw Hill.
10. D. Ray Chaudhuri: "Digital Circuits", vol I & II, 2/e, Platinum Publishers.
11. Tocci and Widmer: "Moss-Digital Systems", 9/e, Pearson Education.
12. J. Bignell and R. Donovan: "Digital Electronics", 5/e, Cenage Learning.

Mathematics - I

Code: MA131
Contacts: 3L
Credits: 3

Module-1 (24 hours)

Probability Theory: Basic Probability Theory, Conditional Probability, Bayes" Theorem, Expected Value and Variance, Distributions – Binomial, Poisson, Normal etc. Statistical Applications: Concepts of mean, median and mode, Objectives of a sample survey, Sample estimate of the attribute of a population, Selection of a sample without bias, Simple random sampling – with and without replacement, Sampling Distributions, Central Limit Theorems, Confidence interval, Hypothesis Testing, Chi-square tests and other testing methods.

Module-2 (12 hours)

Linear Algebra: Matrices and Determinants, Characteristic polynomials, Eigen values, Vector spaces, Concept of Inner Product and Metric, Linear Transformations. Logic and Proofs: Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Inference Rules, Proof Methods and Strategy.

Suggested Books:

1. Introduction to Probability and Statistics by Seymour Lipschutz and John J. Schiller
2. Higher Algebra (Abstract & Linear) by S. K. Mapa
3. Mathematical Probability by A. Banerjee, S. K. De and S. Sen
4. Probability and Stochastic Processes for Engineers by C.W.Helstrom

Elementary Arabic and Islamic Studies

Code: AI131
Contacts: 3L
Credits: 3

Programming in C Lab

Code: BCA191
Contacts: 3P
Credits: 2

Course Objectives:

- To make acquaint the students to know the programming language and also to know how "C" can be used to write serious program to solve the problems.
- Programs will be based on the theoretical paper and to cover the concept of basic arithmetic operations, control statements, arrays, strings, functions, recursions, pointers, structures, unions, file handling, etc.

Course Outcomes:

CO 1: Students should be able to write, compile and debug programs in C language.
CO 2: Understand and use different data types in a computer program.
CO 3: Understand various problem solving methods.
CO 4: Able to design programs involving decision structures, loops and functions.
CO 5: Understand the dynamics of memory by the use of pointers.
CO 6: Able to use different data structures and create/update basic data files.
CO 7: Understand File I/O and Preprocessor directives.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2		√	√			√	
CO3	√	√	√	√	√		√
CO4		√	√	√		√	
CO5		√	√		√		
CO6	√			√	√		√
CO7	√			√	√		

Week 1 Ubuntu and Linux Commands.
Week 2 Designing of flowcharts and algorithms using raptor tool 1. Areas of Polygons. 2. Calculation of Simple and Compound Interest. 3. Swapping of Two numbers with and without temporary variable. 4. Checking whether a number is even or odd. 5. Sum of first 'n' natural numbers. 6. Checking a number whether it is divisible by any given number. 7. Evaluation of mathematical expressions. 8. Programs using scanf( ) and printf( ) statements.
Week 3,4 Programs on operators. (Minimum 4 Programs) Programs on Conditional Statements. (Minimum 4 Programs) Programs on Control Statements. (Minimum 4 Programs)
Week 5 Programs on Functions. (Minimum 6 Programs)
Week 6 Programs on One Dimensional and Two Dimensional Arrays. (Minimum 2 Programs)
Week 7 Programs on Strings with and without string built-in Functions. (Minimum 4 Programs)
Programs on Accessing Structures and Nested Structures. (Minimum 2 Programs)
Week 8 Programs on Array of Structures, Structures and Functions. (Minimum 4 Programs) Programs on Unions, typedef and enum. (Minimum 2 Programs)
Week 9 Programs on Pointers, pointer arithmetic, pointer expression, One Dimensional and Two dimensional arrays. (Minimum 4 Programs)
Week 10 Programs on Pointer to structure, Call by Reference, Pointer to Pointer. (Minimum 3 Programs) Programs on Dynamic Memory Allocation Functions. (Minimum 3 Programs)
Week 11 Programs on Stacks and Queues using Arrays.
Week 12 & 13 Programs on Single Linked List.
Week 14 & 15 Programs on File Operations. (Minimum 6 Programs)
Week 16 Review

Suggested Books:

1. B.W. Kernighan and D.M. Ritchie: The C Programming Language; PHI.
2. E. Balagurusamy: Programming in ANSI C; TMH.
3. Yashwant Kanetkar: Let Us C, BPB Publications, 9th Edition, 2008.
4. B.S. Gottfried: Programming in C; TMH.
5. H. Schildt: C++: The Complete Reference; TMH.
6. B. Stroustrup: The C++ Programming Language; Addison-Wesley.
7. K. N. King: C Programming: A Modern Approach, W. W. Norton and Company.
8. Pradip Dey and Manas Ghosh: Programming in C, Oxford University Press.

PC Software Lab

Code: BCA193
Contacts: 3P
Credits: 2

Course Objectives:

- To define the functions of operating systems along with examples.
- To discuss the different features of word, excel, power point and MS access.
- To discuss basic email features.
- To describe how to browse internet.

Course Outcomes:

CO 1: Demonstrate operating system functions.
CO 2: Create word, excel, power point and ms access files discussing all the features in each of them.
CO 3: Compose and send emails.
CO 4: Browse the internet.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√			√	√	
CO2			√	√	√		√
CO3			√	√	√		
CO4	√		√		√	√	√

Operating System: Familiarization (Keyboard, Memory, I/O Port), Windows, Linux.

Word Processor: Introduction to word, Editing a document, Move and Copy text, Formatting text and Paragraph, Enhancing document, Columns, Tables and Other features.

Introduction to worksheet and shell: Getting started with Excel, Editing cell & using Commands and functions, Moving & Copying , Inserting & Deleting Rows & Columns, Printing work sheet.

Overview of Power point: Basics operation, Animation and Sounds, Presenting shows for corporate and commercial using Power point, Creating charts, Naming ranges and using statistical, math and financial functions, database in a worksheet, Additional formatting commands and drawing toolbar, other commands & functions, multiple worksheet and macros.

Overview of MS Access: Table, Relation, Queries, Reports.

E-mail: Basic Operation, Address Book, Spam and Filtering.

Browsing and Discussion Forum: Browsing and Search, Discussion Forum, Wiki and Google Doc.

Suggested Books:

1. Computer Fundamentals – Raja Raman – Prentice Hall of India 2004.
2. PC Software for Windows 98‟ made simple – R.K.Taxali – Tata McGraw Hill Publishers, 2005.

Digital Logic Lab

Code: BCA195
Contacts: 3P
Credits: 2

Course Objectives:

- To understand the basic concept logic family.
- To know the concepts of Combinational circuits.
- To understand the concepts of flip-flops, registers and counters

Course outcomes:

CO 1: Implement the basic logic gates using diodes.
CO 2: Understand the characteristics of basic logic family.
CO 3: Construct basic combinational circuits and verify their functionalities.
CO 4: Implementation of various Boolean functions.
CO 5: Apply the design procedures to design basic sequential circuits.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√	√			
CO2	√	√		√			
CO3		√	√	√			
CO4		√		√			
CO5		√	√	√			

Logic family: Implementation of OR and AND gates using diodes, Study on characteristics of DTL and TTL inverters using discrete components, Study on characteristics of TTL and CMOS gates. **[12 P]**

Combinational logic circuits: Design and implementation of combinational circuits such as, Adders, comparators, parity generator and checker. Implementation of Boolean functions using multiplexer and decoder/de-multiplexer. **[12 P]**

Sequential circuits: Study of latch and flip-flop, design of counters. **[12 P]**

Suggested Books:

1. Floyed and Jain: "Digital Fundamentals", Pearson Education.
2. Morries Mano: "Digital Logic Design", PHI.
3. Leach & Malvino: "Digital Principles & Application", 5/e, Tata McGraw Hill.
4. Kharate: "Digital Electronics", Oxford.
5. Bigmell and R.Donovan: "Digital Electronics - Logic & Systems", Cambridge Learning.
6. D .J.Comer: "Digital Logic and State Machine Design", 3/e. OUP.
7. P.Raja: "Digital Electronics", Scitech Publications.
8. R.P.Jain: "Modern Digital Electronics", 2/e , Tata McGraw Hill.
9. H.Taub and D.Shilling: "Digital Integrated Electronics", Tata McGraw Hill.
10. D. Ray Chaudhuri: "Digital Circuits", vol I & II, 2/e, Platinum Publishers.
11. Tocci and Widmer: "Moss-Digital Systems", 9/e, Pearson Education.
12. J. Bignell and R. Donovan: "Digital Electronics", 5/e, Cenage Learning.

Semester II:

Data Structures and Algorithms

Code: BCA102
Contacts: 3L
Credits: 3

Course Objectives:

- Identify and use appropriate data structure for a given problem with effective utilization of space and time.
- Describe the linear and nonlinear data structures.
- Analyze the complexities of different sorting techniques.
- Identify trees, recursive functions and Graphs.

Course Outcomes:

CO 1: Get concept of Data Structure and different data types.
CO 2: Understand the time and space complexities of Algorithms.
CO 3: Design a solution to a given problem using arrays.
CO 4: Understand different applications of stacks, queues and linked lists.
CO 5: Choose the appropriate nonlinear data structure and perform operations on them.
CO 6: Understand different sorting techniques and their performances.
CO 7: Develop operations on Binary Search Trees and Graphs.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2		√	√	√			
CO3		√	√	√			
CO4		√	√	√			
CO5		√	√	√			
CO6	√	√		√			
CO7		√	√				

Module - 1 [12 Hrs]

Introduction: Data and Information, Program Structures, Abstract Data Type, Data Structure - Static and Dynamic Data Structures. [3L]
Arrays: Representation of Polynomials and Sparse Matrix, Linear List. [3L]
Linked List: Single Linked Lists, Doubly Linked Lists, Circularly Linked Lists, Linked List Representation of Polynomial and Applications. [6L]

Module - 2 [12 Hrs]

Sorting and Searching Algorithms: Bubble sort, Insertion sort, Selection sort, Merge, Quick, Heap, Radix, Bucket sort, Linear and Binary Search. [6L]
Stack and Queue: Implementations using Arrays and Linked List, Applications, Expression Evaluation and Conversions. [4L]

Recursion: Basic concept, Design of recursive algorithms, Tail recursion. [2L]

Module - 3 [12 Hrs]

Trees: Binary Trees, Binary Search Trees, Height-Balanced And Weight-Balanced Trees, 2-3 Tree, B-Trees, B+ -Trees. Applications of Trees. [6L]
Graphs: Adjacency Matrix and List, Graph Search Algorithms, Spanning Tree Algorithms, Shortest Path Algorithms. [4L]
Hashing: Terminologies, Hashing Functions, Collision Resolution Techniques, Types of Hashing. [2L]

Suggested Books:

1. E. Horowitz, S. Sahni and S. Anderson-Freed: "Fundamentals of Data Structures in C", Second Edition, Universal Press. 2007.
2. M. A. Weiss: "Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2002.
3. A. V. Aho, J. E. Hopcroft and J. D. Ullman: "Data Structues and Algorithms", First Edition, Pearson Education, 2002.
4. R. K. Kruse, Bruce P. Leung: "Data Structures and Program Design", Prentice Hall, 2006.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: "Introduction to Algorithms", Third Edition, PHI Learning Pvt. Ltd, 2010.
6. Y. Langsam, J. M. Augenstein, M. A. Tenenbaum: "Data Structures using C and C++", Second Edition, Pearson Education. 2015.

Numerical Methods

Code: BCA104
Contacts: 3L
Credits: 3

Course Objectives:

- The purpose of this course is to provide basic understanding of the derivation and integration.
- The purpose of this course use of the numerical methods along with the knowledge of finite precision arithmetic.
- To give the knowledge of few interpolation formulas.
- To give some knowledge of linear equations, differential equations etc.

Course Outcomes:

CO 1: Recall the distinctive characteristics of various numerical techniques and the associated error measures.
CO 2: Understand the theoretical workings of various numerical techniques to solve the engineering problems and demonstrate error.
CO 3: Familiar with numerical integration and differentiation, numerical solution of ordinary differential equations.
CO 4: Familiar with numerical solutions of nonlinear equations in a single variable.
CO 5: Apply the principles of various numerical techniques to solve various problems.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2		√		√			
CO3				√			
CO4		√		√			
CO5				√			

Module-1 (12 hours)

Approximation in numerical computation: Truncation and rounding errors, Fixed and floating-point arithmetic, Propagation of errors. Interpolation: Newton forward & backward interpolation, Lagrange‟s and Newton‟s divided difference Interpolation.

Module-2 (12 hours)

Numerical solution of a system of linear equations: Gauss elimination method, Matrix inversion, LU Factorization method, Gauss-Jacobi and GaussSeidel iterative methods.
Numerical solution of Algebraic equation: Bisection method, Secant method, Regula-Falsi method, Newton-Raphson method.

Module-3 (12 hours)

Numerical solution of ordinary differential equation: Taylor‟s series method, Euler‟s method, Runge Kutta methods, Predictor-Corrector methods and Finite Difference method.
Numerical integration: Trapezoidal rule, Simpson‟s 1/3 rule, Weddle‟s rule.

Suggested Readings:

1. C.Xavier: C Language and Numerical Methods.
2. Dutta & Jana: Introductory Numerical Analysis.
3. J.B.Scarborough: Numerical Mathematical Analysis.
4. Jain, Iyengar , & Jain: Numerical Methods (Problems and Solution).

System Analysis and Design

Code: BCA106
Contacts: 3L
Credits: 3

Course Objectives:

- To provide a basic exposure to the fundamental concepts of system analysis and design.
- To familiarize with the information requirement analysis.
- To provide the essentials for system design.
- To explain different types of testing to understand realistic problems.

Course Outcomes:

CO 1: Students will be able to understand the basic stuffs of system analysis and design.
CO 2: Students will be competent to gather data for analysis and specify the requirements of a system.
CO 3: Students will be proficient to design a system.
CO 4: Students will be able to learn about different types of software testing.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2	√	√		√	√	√	
CO3	√	√	√	√	√	√	
CO4	√	√	√	√	√	√	

Module 1 (12 hrs)

Overview of System analysis and design: Development life cycle (Waterfall, Spiral, incremental models), feasibility studies, Requirements determination, Logical design, Physical design, Program design, Risk and feasibility analysis, prototyping

Module 2 (12 hrs)

Information requirement analysis: Process modelling with physical and logical data flow diagrams, Data modelling with entity relationship diagrams, Normalization upto 3NF

Module 3 (12 hrs)

System design: Process descriptions, Input/output controls, object modelling, Database design, User Interface design, Documentation, Data Dictionary, Development methodologies: Top down, bottom up, structured chart, decision table, decision tree, CASE productivity tools.

Module 4 (12 Hrs)

Testing – Unit, integration, system, Acceptance, regression, Test Case generation, Case studies

Suggested Books:

1. Parthasarathi: "System Analysis & Design", EPH.
2. Rajaraman: "Analysis & Design of Information Systems", PHI
3. Senn: "Analysis & Design of Information Systems", MH
4. Ram Bansal, Vigyacharya: "Information Systems: Analysis and Design", New Age International.
5. "System Analysis, Design & MIS", EXCEL BOOKS.
6. Sharma: "Analysis, Design & Implementation of Information System", VIKAS.
7. V.K. Jain: "System Analysis & Design", Wiley Dreamtech.

Mathematics - II

Code: MA132
Contacts: 3L
Credits: 3

Module-1 (8 hours)
Sets, Functions, Relations: Sets and Set Operations, Cartesian Product, Functions – one-one, onto, one-to-one, Relations, Equivalence Relation and Partitions, Partial Order Relations, Lattices.

Module-2 (12 hours)
Abstract Algebra: Binary Operations, Groupoid, Semi-group and Monoid, Group and Subgroup, Cosets, Lagrange‟s theorem, Cyclic group, Order of a group, Generators, Normal subgroup, Quotient group, Homomorphism, Isomorphism, Permutation group, Direct product, Rings and sub-rings, Ideals and quotient rings, Integral domains and Fields.

Module-3 (16 hours)
Introduction to Number Theory: Divisibility, GCD, Prime Numbers, Infinitude of Primes, Fundamental Theorem of Arithmetic, Congruences, Fermat‟s Little Theorem, Euler‟s Formula, Chinese Remainder Theorem. Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction Basic Counting: Sum Rule and Product Rule, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Generalized Permutations and Combinations

Suggested Readings:

1. Kenneth H. Rosen, “Discrete Mathematics and its Applications”, McGraw Hill, 2002.
2. J.P.Tremblay & R. Manohar, “Discrete Mathematical Structure with Applications to Computer Science” Mc.Graw Hill, 1975.
3. V. Krishnamurthy, “Combinatories:Theory and Applications”, East-West Press.
4. Seymour Lipschutz, M.Lipson, “Discrete Mathemataics” Tata McGraw Hill, 2005.

Common English

Code: EN132
Contacts: 3L
Credits: 3

Data Structures and Algorithms Lab

Code: BCA192
Contacts: 3P
Credits: 2

Course Objectives:

- Design and analyze linear and nonlinear data structures.
- Acquire programming skills to implement sorting and searching techniques.
- Identify and apply the suitable data structure for the given real world problem.

Course Outcomes:

CO 1: Implement different operations like insert, delete, search, sort and traverse using arrays.

CO 2: Implementation of stacks and queues using linked lists.
CO 3: Apply recursive and non-recursive traversal of trees.
CO 4: Implement sorting techniques.
CO 5: Implement hashing techniques

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2		√	√	√			
CO3		√	√	√			
CO4		√	√	√	√		
CO5		√	√	√	√		

Experiments should include but not limited to:
Implementation of stacks and queues using arrays.
Implementation of stacks and queues using linked lists: inserting, deleting, and inverting a linked list.
Applications of linked lists: polynomial arithmetic, set operations, etc.
Sparse Matrices: Multiplication, addition.
Recursive and Non-recursive traversal of trees.
Hash tables implementation: searching, inserting and deleting, searching & sorting techniques.

Suggested Books:

1. E. Horowitz, S. Sahni and S. Anderson-Freed:"Fundamentals of Data Structures in C", Second Edition, Universal Press. 2007.
2. M. A. Weiss:"Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2002.
3. R. K. Kruse, Bruce P. Leung:"Data Structures and Program Design", Prentice Hall, 2006.
4. Y. Langsam, J. M. Augenstein, M. A. Tenenbaum:"Data Structures using C and C++", Second Edition, Pearson Education. 2015.

Numerical Methods LAB

Code: BCA194
Contacts: 3P
Credits: 2

Course Objectives:

- To provide suitable and effective methods called Numerical Methods, for obtaining approximate representative numerical results of the problems.
- To solve problems in the field of Applied Mathematics, Theoretical Physics and Engineering which requires computing of numerical results using certain raw data?
- To solve complex mathematical problems using only simple arithmetic operations. The approach involves formulation of arithmetical models of physical situations that can be solved with arithmetic operations.

- To deal with various topics like finding roots of equations, solving systems of linear algebraic equations, interpolation and regression analysis, numerical integration & differentiation, solution of differential equation, boundary value problems, and solution of matrix problems.
- To facilitate numerical computing.

Course Outcomes:

CO 1: Apply the programming skills to solve the problems using multiple numerical approaches and statistics.
CO 2: Analyze the results to design reports by effective presentation.
CO 3: Familiar with finite precision computation.
CO 4: Familiar with numerical solutions of nonlinear equations in a single variable.
CO 5: Familiar with numerical integration and differentiation, numerical solution of ordinary differential equations.
CO 6: Familiar with calculation and interpretation of errors in numerical method.

CO-PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√					
CO2			√				
CO3			√	√			
CO4			√	√			
CO5		√		√			
CO6							

Numerical Methods LAB

1. Assignments on Newton forward /backward, Lagrange's interpolation.
2. Assignments on numerical integration using Trapezoidal rule, Simpson's 1/3 rule, Weddle's rule.
3. Assignments on numerical solution of a system of linear equations using Gauss elimination and Gauss-Seidel iterations.
4. Assignments on numerical solution of Algebraic Equation by Regular-falsi and Newton Raphson methods.
5. Assignments on ordinary differential equation: Euler's and Runga-Kutta methods.
6. Introduction to Software Packages: Matlab / Scilab / Labview / Mathematica

Suggested Books:

1. C.Xavier: C Language and Numerical Methods.
2. Dutta & Jana: Introductory Numerical Analysis.
3. N. Dutta: Computer Programming & Numerical Analysis, Universities Press.
4. Srimanta Pal: Numerical Methods, OUP.
5. J.B. Scarborough: Numerical Mathematical Analysis, Oxford & IBH Pub. Co. Pvt. Ltd.
6. S.C. Chapra and R.P. Canale: Numerical Methods for Engineers, Tata McGraw-Hill.
7. Balagurusamy: Numerical Methods, Scitech.
8. Baburam: Numerical Methods, Pearson Education.
9. Soumen Guha & Rajesh Srivastava: Numerical Methods, OUP.

Communicative English Lab

Code: HN198
Contacts: 3P
Credits: 2

Semester III

Formal Language and Automata Theory

Code: BCA201
Contacts: 3L
Credits: 3

Course Objectives:

- Course should provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (and less magical) view towards algorithmic design and in general computation itself.
- The course should in addition clarify the practical view towards the applications of these ideas in computer science.

Course Outcomes:

CO 1: Get knowledge about different computational models and analyse them using combinatorial methods.
CO 2: Apply formal mathematical methods to prove properties of languages, grammars.
CO 3: Understand algorithms for different problems.
CO 4: Identify limitations of some computational models and possible methods of proving them.
CO 5: Get an overview on the theoretical study in this course.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2		√	√	√			
CO3		√	√	√			
CO4		√		√			
CO5		√	√	√	√		

Module - 1 [12 Hrs]
Introduction: Alphabet, Languages, Grammar, Chomsky Hierarchy of Languages, Regular Expressions and Languages. [4L]
Finite Automata (FA): Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Deterministic Finite Automata (DFA) and Regular Expressions, Nondeterministic Finite Automata (NFA) and Equivalence with DFA, Regular Grammars and Equivalence with Finite Automata, Properties of Regular Languages, Pumping Lemma for Regular Languages, Minimization of Finite Automata. [8L]

Module - 2 [12 Hrs]

Context-Free Languages and Pushdown Automata: Context-Free Grammars (CFG) and Languages (CFL), Pushdown Automata (PDA) and Equivalence with CFG, Parse Trees, Ambiguity in CFG, Pumping Lemma for Context-Free Languages, Deterministic Pushdown Automata, Closure Properties of CFLs. [6L]

Context-Sensitive Languages: Context-Sensitive Grammars (CSG) and Languages, Linear Bounded Automata and Equivalence with CSG. [6L]

Module - 3: [12 Hrs]

Turing Machines: The Basic Model for Turing Machines (TM), Turing-Recognizable (Recursively Enumerable) and Turing-Decidable (Recursive) Languages and Their Closure Properties, Variants of Turing Machines, Nondeterministic TMs and Equivalence with Deterministic TMs, Universal TMs, Halting Problem, Recursive Functions and Sets, Recursively Enumerable Sets. [6L]

Undecidability: The Church-Turing Thesis, Universal Turing Machine, The Halting Problem and Diagonalization Languages, Reduction Between Languages and Rice's Theorem, Undecidable Problems about Languages. [6L]

Suggested Books:

1. John Martin: "Introduction to Languages and Theory of Computation",
2. J. E. Hopcroft and J. D. Ullman: "Introduction to Automata Theory, Languages and Computation",
3. H. R. Lewis and C. H. Papadimitriou: "Elements of the Theory of Computation",
4. Peter Linz, Narosa: "An Introduction to Formal Languages and Automata",
5. Michael Sipser: "Introduction to the Theory of Computation", Thomson Press.
6. Dexter C. Kozen: "Automata and Computability", Springer.

Computer Organization and Architecture

Code: BCA203

Contacts: 3L

Credits: 3

Course Objectives:

- To familiar with Basic Structure of Computer Systems.
- To differentiate hardware and software.
- To learn the fundamentals of pipelining.
- To learn memory and data storage mechanism.
- To acquire knowledge about peripherals.

Course Outcomes:

CO 1: Knowledge about the fundamental organization of a computer system.

CO 2: Understand how integer and real numbers are represented and instruction is

getting executed.

CO 3: Explain addressing modes, instruction formats and program control statements.

CO 4: Understand memory organization and their operation principles.

CO 5: Familiarity with peripheral devices and different data transfer techniques.

CO 6: Knowledge about fundamentals concepts of pipelining and its hurdles.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√		√		√		
CO2	√		√				
CO3	√		√	√	√		
CO4	√	√		√	√		
CO5	√		√				
CO6	√		√				

Unit I [10 Hrs]

Functional units-Basic operational concepts-Bus structure-Performance and metrics- Instruction and its sequence -Hardware and software interface-Instruction set architecture - Addressing modes-RISC - CISC -ALU design-Fixed point and Floating point operation.

Unit II [9 Hrs]

Fundamental Concepts – Execution of complete instruction – Multiple bus organization – Hardwired control – Microprogrammed control – Nano programming.

Unit III [10 Hrs]

Pipelining Basic concepts – Data hazards – Instruction hazards – Influence on instruction sets – Data path and control considerations – Performance considerations – Exception handling.

Unit IV [9 Hrs]

Memory, Basic concepts – Semiconductor RAM – ROM – Speed – Size and Cost – Cache memories – Improving cache performance – Virtual memory – Memory management requirements – Associative memories – Secondary storage devices

Unit V [10 Hrs]

Accessing I/O Devices – Programmed I/O– Interrupts – Direct memory access – Buses – Interface circuits – Standard I/O interfaces (PCI, SCSI, USB) – I/O devices and processors.

Suggested Books:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky: “Computer Organization”, 5th Edition, Tata Mc-Graw Hill, 2002.
2. Heuring, V.P. and Jordan, H.F.: “Computer Systems Design and Architecture”, 2nd Edition, Pearson Education, 2004.
3. Patterson, D. A., and Hennessy, J.L.: “Computer Organization and Design: The Hardware/Software Interface”, 3rd Edition, Elsevier, 2005.
4. William Stallings: “Computer Organization and Architecture – Designing for Performance”, 6th

Edition, Pearson Education, 2003.
5. Hayes, J.P.: "Computer Architecture and Organization", 3rd Edition, Tata Mc-Graw Hill, 1998.

Operating Systems

Code: BCA205

Contracts: 3L

Credits: 3

Course Objectives:

- To understand the main components and different functions of an operating system.
- To understand about process and its states, synchronized and different scheduling methods.
- To compare and illustrate various process scheduling algorithms.
- To understand various issues in Inter Process Communication (IPC) and the role of OS in IPC
- To understand different approaches to memory management.
- To summarize the principles of Virtual memory as applied to paging & caching techniques.
- To demonstrate internal file system structure with device drivers and file operations using system calls.
- To study the need for special purpose operating system with the advent of new emerging technologies

Course Outcomes:

CO 1: Understand the working procedure of an operating system and its components.
CO 2: Describe process management techniques and analyze the synchronization methods.
CO 3: Identify the working methodology of multithreaded applications and distinguish different scheduling algorithms.
CO 4: Evaluate the requirement for process synchronization and coordination handled by operating system.
CO 5: Identify the reasons of deadlocks, and their remedial measures in an operating system.
CO 6: Understand different memory management techniques used in operating systems.
CO 7: Get overview of different file systems.

CO-PO mapping

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√					
CO2		√	√				
CO3		√		√			
CO4			√				√
CO5				√			
CO6		√	√				
CO7			√	√			

Module 1: [12 Hrs]

Introduction to Operating Systems.

Concept of batch-processing, multi-programming, time sharing, real time operations.

Process Management: Concept of process, state diagram, process control block; scheduling of processes – criteria, types of scheduling, non-preemptive and preemptive scheduling algorithms like: FCFS, Shortest Job First/Next (SJF/N), Shortest Remaining Time Next (SRTN), Round Robin (RR), Highest Response ratio Next (HRN), Priority based scheduling, different Multilevel queue scheduling etc.

Threads – concept, process vs thread, kernel and user threads, multithreading models.

Inter-process Communication (IPC) – Shared memory, message, FIFO, concept of semaphore, critical region, monitor.

Module 2: [12 Hrs]

Process Synchronization: concepts, race condition, critical section problem and its solutions; synchronization tools- semaphore, monitor etc., discussion of synchronization problems like producer-consumer, readers-writers, dining philosophers, sleeping-barber etc.

Deadlock – conditions, resource allocation graph, prevention techniques, avoidance technique – Banker‟s algorithm and related algorithms.

Module 3: [12 Hrs]

Memory management: Address space and address translation; static partitioning, dynamic partitioning, different types of fragmentation, paging, segmentation, swapping, virtual memory, demand paging, page size, page table, page replacement algorithms – FIFO, LRU, Optimal page replacement, Variants of LRU, etc; thrashing, working set strategy.

Module 4: [12 Hrs]

File Management: File and operations on it, file organization and access; file allocation; directory structures, file sharing, file protection

Device management: Magnetic disks, disk scheduling- criteria, algorithms – FCFS, SSTF, SCAN, C-SCAN, LOOK, etc, disk management – formatting, boot block, disk free space management techniques, concept of RAID etc.

Protection and Security: Concepts of domain, Access matrix and its implementation, access control, Security of systems- concepts, threats- Trojan horse, virus, worms etc, introduction to cryptography as security tool, user authentication.

Suggested Books:

1. A. Silberschatz, P. Galvin and G. Gagne: “Operating Systems Concepts”, Wiley India.
2. Gary Nutt, N. Chaki and S. Neogy: “Operating Systems Concepts”, Pearson Education.
3. W. Stallings: “Operating Systems”, Pearson Education.
4. D. M. Dhamdhere: “Operating Systems: A Concept-based Approach”, Tata McGraw-Hill.

Object Oriented Programming

Code: BCA207

Contacts: 3L
Credits: 3

Course Objectives:

- To understand the fundamental concepts of object oriented programming paradigm.
- To learn the benefits of object oriented programming technique.
- To understand of writing object oriented programming using Java language.
- To learn the fundamental concept about packages, multithreading etc.
- To learn designing applications for solving real life problems using object oriented programming.

Course Outcomes:

CO 1: Learning object oriented programming paradigm in designing application software.
CO 2: Learn the difference among structured and object oriented programming paradigm to design a solution for real life problems.
CO 3: Understanding basic object oriented principles like inheritance, encapsulation and polymorphism to solve real life computing problems using Java language.
CO 4: Learn code reusability through inheritance, packages and interface.
CO 5: Learn to develop multithreaded programs, packages, exception handling in Java.
CO 6: Learn to write Applet programs.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√	√	√		
CO2	√	√	√	√	√		
CO3		√	√	√	√	√	
CO4			√	√		√	
CO5		√	√	√	√		
CO6			√	√		√	
CO7	√	√	√	√	√		

UNIT I: Object Oriented Thinking [3 Hrs]

Need for OOP Paradigm, Principles of Object Oriented Languages, Benefits of OOP, Applications of OOP.

UNIT II: Java Basics [8 Hrs]

History of Java, Java Buzzwords, Java Virtual Machine, Platform Independence, Data Types, Variables, Scope and Life time of variables, Operators, Expressions, Control Statements, Type Conversion and Casting, Simple Java Program.

UNIT III: Classes and Objects [8 Hrs]

Concepts of Classes, Objects, methods, constructors, this keyword, garbage collection, Compile time polymorphism: overloading methods and constructors, parameter passing, command line arguments, Recursion, nested and inner classes, Exploring String, StringBuffer classes, Arrays.

UNIT III: Inheritance [6 Hrs]

Hierarchical abstractions, Base class object, subclass, subtype, forms of inheritance, benefits of inheritance, Member access rules, Usage of super, static and final with inheritance, Run time polymorphism: method overriding, abstract classes, the Object class.

UNIT IV: Packages and Interfaces [5 Hrs]

Defining, Creating and Accessing a Package, Understanding CLASSPATH, access control, differences between classes and interfaces, defining an interface, implementing interface, variables in interface and extending interfaces. Exploring java.io.

UNIT V: Exception Handling [5 Hrs]

Concepts of exception handling, benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception classes: throw and throws.

UNIT VI: Multithreading [8 Hrs]

Differences between multi-threading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads.

UNIT VII: Applet Programming [5 Hrs]

Applet & Application, Applet Architecture, Parameters to Applet, Embedding Applets in Web page, Applet Security Policies.

Suggested Books:

1. E Balagurusamy: "Programming with Java", McGraw Hill Education.
2. Herbert Schildt: "Java: The Complete Reference", McGraw Hill Education.
3. Sachin Malhotra and Saurabh Choudhary: "Programming in Java", Oxford University Press.
4. Y. Daniel Liang: "Introduction to Java Programming, Brief Version", Pearson Education.
5. Y. Daniel Liang: "Introduction to Java Programming, Comprehensive Version", Pearson Education.
6. Cay S. Horstmann: "Core Java - Vol. I and Vol. II", Pearson Education.
7. E. Balagurusamy: "Object-Oriented Programming with C++", McGraw Hill Education.
8. Bjarne Stroustrup: "The C++ Programming Language", Pearson Education.
9. R. Lafore: "Object Oriented Programming in C++", Pearson Education.
10. Debasish Jana: "C++ and Object-Oriented Programming Paradigm", PHI Learning.

OOPS Lab

Code: BCA291
Contracts: 3P
Credits: 2

Course Objectives:

- Learn fundamental programming constructs like variables, conditional and iterative statements, methods, etc.
- Learn fundamentals of object-oriented programming in Java: defining classes, invoking methods, use of Java class libraries, etc.
- Learn to use the Java SDK environment for creating, debugging and running simple Java programs.
- To learn principles of software development thorough Java language.
- Ability for writing application programs to solve real life problems through Java.

Course Outcomes:

CO 1: Learn to implement object orientated programs in Java through class – object – constructor relationship.

CO 2: Fundamental knowledge about code reusability in Object Oriented Programming.

CO 3: Understanding the usage of various keywords for encapsulation and polymorphism techniques in object oriented programming.

CO 4: Fundamental knowledge about data abstractions via inheritance, interface and packages.

CO 5: Learn to use arrays, Strings and wrapper classes in real life problems.

CO 6: Learn to write multithreading, exception handling and applet programs.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√	√	√		
CO2	√	√	√	√	√	√	
CO3		√	√	√	√	√	
CO4		√	√	√	√	√	
CO5	√	√	√	√	√		
CO6	√	√	√	√	√	√	

1. Assignments on class, constructor, overloading, inheritance, overriding
2. Assignments on abstract classes, String handling
3. Assignments on wrapper class, arrays
4. Assignments on developing interfaces- multiple inheritance, extending interfaces
5. Assignments on creating and accessing packages
6. Assignments on Exception handling
7. Assignments on multithreaded programming
8. Assignments on applet programming

Suggested Books:

1. E Balagurusamy: “Programming with Java”, McGraw Hill Education
2. Herbert Schildt:“Java: The Complete Reference”, McGraw Hill Education
3. Sachin Malhotra and Saurabh Choudhary: “Programming in Java”, Oxford University Press
4. Y. Daniel Liang: “Introduction to Java Programming, Brief Version”, Pearson Education
5. Y. Daniel Liang:“Introduction to Java Programming, Comprehensive Version”, Pearson Education
6. Cay S. Horstmann: “Core Java - Vol. I, Vol. II and Vol. II”, Pearson Education
7. E. Balagurusamy: “Object-Oriented Programming with C++”, McGraw Hill Education

8. Bjarne Stroustrup: "The C++ Programming Language", Pearson Education
9. R. Lafore: "Object Oriented Programming in C++", Pearson Education
10. Debasish Jana: "C++ and Object-Oriented Programming Paradigm", PHI Learning

Computer Architecture Lab

Code: BCA293
Contracts: 3P
Credits: 2

Course objectives:

- Familiarity with the principles of combinational circuits.
- Understand the design of sequential digital logic circuits.
- Techniques to optimize at gate level for designing various circuits.
- Familiarity with ICs and simulate different digital circuit using tools.

Course outcomes:

CO 1: Familiarity with software tools and different basic hardware components.
CO 2: Learn to design various circuits using ICs.
CO 3: Understand basic structure of different digital components- multiplexer, decoder, encoder etc.
CO 4: Capable of designing ALU and CPU.
CO 5: Understand interfacing of memory with CPU.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√				√		
CO2	√		√		√		√
CO3	√				√		
CO4	√		√		√		√
CO5	√		√		√		

All laboratory assignments are based on Hardware Description Language (VHDL or Verilog) Simulation. [Pre-requisite: The hardware based design has been done in the Digital logic laboratory and Computer Organization Laboratory]

1. HDL introduction
2. Basic digital logic base programming with HDL
3. 8-bit Addition, Multiplication, Division
4. 8-bit Register design
5. Memory unit design and perform memory operatons.
6. 8-bit simple ALU design
7. 8-bit simple CPU design

8. Interfacing of CPU and Memory

Suggested Books:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky: "Computer Organization", 5th Edition, Tata Mc-Graw Hill, 2002.
2. Heuring, V.P. and Jordan, H.F.: "Computer Systems Design and Architecture", 2nd Edition, Pearson Education, 2004.
3. Patterson, D. A., and Hennessy, J.L.: "Computer Organization and Design: The Hardware/Software Interface", 3rd Edition, Elsevier, 2005.
4. William Stallings: "Computer Organization and Architecture – Designing for Performance", 6th Edition, Pearson Education, 2003.
5. Hayes, J.P.: "Computer Architecture and Organization", 3rd Edition, Tata Mc-Graw Hill, 1998.

Operating Systems Lab

Code: BCA295
Contracts: 3P
Credits: 2

Course Objectives:

- To discuss basic UNIX commands.
- To write and demonstrate Shell programming.
- To understand process creation, execution, deletion and different types of processes in the system.
- To write semaphore and threads programming.
- To understand inter-process communication.

Course Outcomes:

CO 1: Understand basic UNIX commands.
CO 2: Able to write and develop Shell programming.
CO 3: Understand process creation, execution, deletion and different types of processes in the system.
CO 4: Able to write semaphore and threads programming.
CO 5: Able to understand inter-process communication.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1			√				
CO2		√	√	√			
CO3				√			
CO4		√	√	√			
CO5		√	√	√			

Shell programming: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).
Process: starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.
Signal: signal handling, sending signals, signal interface, signal sets.
Semaphore: programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).
POSIX Threads: programming with pthread functions(viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel)
Inter-process communication: pipes(use functions pipe, popen, pclose), named pipes(FIFOs, accessing FIFO)

Suggested Books:

1. UNIX concepts and Applications, Sumitabha Das, 4th Edition, Tata McGraw-Hill
2. Introduction to UNIX SHELL Programming, M.G. Venkateshmurthy, Pearson Education
3. UNIX and SHELL Programming, B.M. Harwani, Oxford Higher Education

Semester IV:

Database Management System

Code: BCA202
Contacts: 3L
Credits: 3

Course Objectives:

- The objective of this course is
- Students should be exposed to the fundamental concepts of Data Base Management Systems
- Students should be able to model, design and implement Data Base Management Systems
- Students should be able to manipulate a database using Structured Query Language (SQL)
- Students should be able to master techniques for Database Normalization, Transaction Management and database security and recovery management.

Course Outcomes:

CO 1: Understanding functional components and architecture of DBMS and its role in Information System
CO 2: Mastering SQL queries with background understanding of Relational Algebra
CO 3: Applying E-R model and Relational model for designing and implementation of DBMS
CO 4: Understanding Integrity Constraint, Functional Dependency and Normalization Rules
CO 5: Understanding Transaction Processing & Concurrency control
CO 6: Understanding Query Optimization, indexing, storage and recovery management

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7

CO1		√					
CO2		√	√		√		
CO3		√		√	√	√	
CO4	√	√		√			
CO5				√	√	√	
CO6	√	√					

Introduction [6 Hrs]

Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema Architecture of DBMS.

Entity-Relationship Model [5 Hrs]

Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features.

Relational Model [7 Hrs]

Structure of relational Databases, Relational Algebra, Relational Calculus, Extended Relational Algebra Operations, Views, Modifications of the Database.

SQL and Integrity Constraints [6 Hrs]

Concept of DDL, DML, DCL. Basic Structure, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, Views, Nested Subqueries, Database security application development using SQL, Stored Procedures and Triggers.

Relational Database Design [9 Hrs]

Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, 2NF, 3NF, Boyce-Codd Normal Form, Normalization using multi-valued dependencies, 4NF, 5NF, Lossless Decomposition

Internals of RDBMS [8 Hrs]

Physical data structures, Query optimization: join algorithm, statistics and cost based optimization. Transaction Processing, Concurrency Control and Recovery Management, Serializability, Lock based protocols, Two Phase Locking.

File Organization & Index Structures [7 Hrs]

File & Record Concept, Placing file records on Disk, Fixed and Variable sized Records, Types of Single-Level Index (primary, secondary, clustering), Multilevel Indexes, Dynamic Multilevel Indexes using B tree and B+ tree.

Suggested Books:

1. Abraham Silberschatz, Henry F. Korth and S Sudarshan: “Database System Concepts”, McGraw Hill Education.
2. Elmasri Ramez and Novathe Shamkant: “Fundamentals of Database Systems”, Pearson Education.
3. Raghu Ramakrishnan and Johannes Gehrke: “Database Management Systems”, McGraw Hill Education.
4. Jim Gray and Andreas Reuter: “Transaction Processing: Concepts and Techniques”, Moragan Kauffman Publishers.

5. C.J. Date: “An Introduction to Database Systems”, Pearson Education.
6. R. Panneerselvam: “Database Management Systems”, PHI Learning.
7. Alexis Leon and Mathews Leon: “Fundamentals of Database Management Systems”, McGraw Hill Education.
8. Ullman JD.: “Principles of Database Systems”, Galgotia Publications.

Computer Graphics

Code: BCA204
Contacts: 3L
Credits: 3

Course Objectives:

- To introduce computer graphics concepts, display systems, algorithm design, elemental algorithms and transformations.
- To make students familiar with techniques of object representation and viewing in two dimensional and three dimensional space.
- To learn about elemental concepts of modeling, rendering and shading and their applications.

Course Outcomes:

CO 1: Knowledge about fundamentals of computer graphics.
CO 2: Concepts of algorithm design and geometric/mathematical transforms.
CO 3: Understanding of two dimensional object representation and viewing in display devices.
CO 4: Knowledge about representation, transformation and viewing objects in three dimensional space.
CO 5: Familiarity with different aspects and components about recent applications of computer graphics and insights into upcoming prospective graphics-powered applications.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√		√	√		√
CO2	√	√		√	√		√
CO3	√	√		√	√	√	√
CO4	√	√		√	√	√	√
CO5	√	√		√	√	√	

Unit I. Introduction to Computer Graphics & Graphics Systems (8L)

Overview of computer graphics, graphical display devices, representing pictures, preparing, presenting & interacting with pictures for presentations; Visualization & image processing; RGB color model, direct coding, lookup table; storage tube graphics display, Raster scan display, 3D viewing devices, Plotters, printers, digitizers, Light pens etc.; Active & Passive graphics devices; Computer graphics softwares; Character generation

Unit II. Scan Conversion (8L)

Points and lines, Line drawing algorithms; DDA algorithm, Bresenham‟s line algorithm, Circle generation algorithm; Ellipse generating algorithm; scan line polygon fill algorithm, boundary fill

algorithm, flood fill algorithm.

Unit III. 2D Transformation (6L)

Basic transformations - translation, rotation, scaling; Matrix representations & homogeneous coordinates, transformations between coordinate systems; reflection, shear, Transformation of points, lines, parallel lines, intersecting lines.

Unit IV. Viewing in 2D (6L)

Viewing pipeline, Window to viewport co-ordinate transformation, clipping operations, point clipping, line clipping, Cohen Sutherland Algorithm, clipping circles, polygons & ellipse, Sutherland Hodgeman algorithm.

Unit V. 3D Transformation and Viewing (6L)

Translation, rotation, scaling & other transformations, Rotation about an arbitrary axis in space, reflection through an arbitrary plane; general parallel projection transformation; clipping, viewport clipping, 3D viewing.

Unit VI. Curves and Surfaces (4L)

Object representation; Curve and surface designs, Bezier curves, Continuity conditions; B-spline curves, end conditions for periodic B-spline curves, rational B-spline curves.

Suggested Books:

1. Computer Graphics C version, by D. Hearn and M.P. Baker, Pearson Education.
2. Computer Graphics bySamit Bhattacharya, Oxford University Press.
3. Procedural Elements for Computer Graphics by David F.Rogers, TMHpublication.
4. Mathematical Elements for Computer Graphics by David F. Rogers and J. A.Adams, TMH publication.
5. Computer Graphics Principles &Practice in C by J.D. Foley, A. van Dam, S.K.Feiner and F. H. John, Pearson.
6. Computer Graphics – A Programming Approach by S. Harrington, TMHpublication.
7. Computer Graphics by A.N. Sinha and A.D. Udai, TMH publication

Computer Networks

Code: BCA206
Contracts: 3L
Credits: 3

Course Objectives:

- Study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- Read the fundamentals and basics of Physical layer, and will apply them in real time applications
- Study data link layer concepts, design issues, and protocols.
- Gain core knowledge of Network layer routing protocols and IP addressing.
- Study Transport layer services and protocols.
- Acquire knowledge of Application layer and Presentation layer paradigms and protocols.

Course Outcomes:

CO 1: Understand basic concepts of computer network and its components.
CO 2: Describe the functions of each layer in OSI and TCP/IP model.
CO 3: Explain the types of transmission media.
CO 4: Describe the functions of data link layer and explain the protocols.
CO 5: Classify the routing protocols and analyze how to assign the IP addresses for the given network
CO 6: Describe the functions of Transport layer and explain the protocols.
CO 7: Explain the functions of Application layer.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√		√			
CO2		√	√	√			
CO3		√		√			
CO4		√	√	√			
CO5		√	√	√			
CO6		√	√				
CO7		√	√	√			

Module 1: [12 Hrs]

Introduction: Uses of Computer Networks, Types of Computer Networks, OSI Reference Model, Example Networks

Physical Layer: Data and signal fundamentals, Transmission impairments, Attenuation, Distortion, Noise, Data rate limits for noisy and noiseless channels, Performance

Digital Transmission – Problems with digital transmission, Different line coding schemes, Block coding schemes, Scrambling techniques, Analog to digital encoding, Analog Transmission.

Transmission Media - Guided (wired) media – Twisted pair cable, Coaxial cable and Fibre optic cable, Unguided (wireless) media – Different propagation modes, Radio waves, Terrestrial microwaves, Satellite communication. Concept of multiplexing, Frequency division multiplexing, Time division multiplexing – Synchronous and Statistical time division multiplexing, Handling variable length data, Pulse stuffing. Concept of spreading spectrum, Frequency hopping spread spectrum and Direct sequence spread spectrum.

Module 2: [12 Hrs]

Data Link Layer: Link Layer Services, Error detection and Correction Techniques, Multi Access Protocols, Link Layer Addressing, Ethernet, Hubs, Switches and Switches, Point to Point Protocol, Asynchronous Transfer Mode, Multiprotocol Label Switching

Module 3: [12 Hrs]

Network Layer: Introduction, Virtual Circuit and Datagram Networks, IP Addressing, Subnetting, Routing Algorithms (Link State, Distance Vector, Hierarchical), Routing in the Internet (RIP, OSPF, BGP), Broadcast and Multicast Routing Algorithms, Routers, ICMP, IPv6

Module 4: [12 Hrs]

Transport Layer: Introduction to Transport Layer Services, Connectionless Transport: UDP,

Principles of Reliable Data Transfer, Connection Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control, Sockets, Quality of services (QOS)
Application Layer: Web and HTTP, Domain Name Space (DNS), Electronic Mail (SMTP, MIME, IMAP, POP3), File Transfer Protocol, Cryptography.

Suggested Books:

1. James F. Kurose and Keith W. Ross: "Computer Networking: A Top-Down Approach Featuring the Internet", 5th Edition, Pearson Education, 2010.
2. Behrouz A. Forouzan: "Data communication and Networking", 4th Edition, Tata McGraw-Hill, 2007.
3. Andrew S. Tanenbaum: "Computer Networks", 4th Edition, Prentice Hall India, 2003.
4. Larry L. Peterson and Peter S. Davie: "Computer Networks: A Systems Approach", 4th Edition, Morgan Kauffman Publishers, 2007.
5. William Stallings: "Data and Computer Communication", 9th Edition, Pearson Education, 2011.

Design and Analysis of Algorithms

Code: BCA208
Contacts: 3L
Credits: 3

Course Objectives:

- Define the basic concepts of algorithms and analyze the performance of algorithms.
- Discuss various algorithm design techniques for developing algorithms.
- Discuss various searching, sorting and graph traversal algorithms.
- Understand NP completeness and identify different NP complete problems.
- Discuss various advanced topics on algorithms.

Course Outcomes:

CO 1: Student will be able to get idea of algorithms and its running time complexity.
CO 2: Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
CO 3: Student will be able to solve explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
CO 4: Student will learn about computational complexity of problems & can use this in practical problem solving.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√	√				

CO2	√			√			
CO3		√		√			
CO4			√	√			

Module - 1 [12 Hrs]
Models of computation: RAM, TM etc. time and space complexity.

Asymptotic Notation: Big-O, omega, theta etc.; finding time complexity of well known algorithms like heap sort, search algorithm etc.
Algorithm design techniques:. Recursion: use, limitations, examples.

Divide and Conquer: basic concept, use, examples (Merge sort, Quick Sort, Binary Search).

Module - 2 [12 Hrs]
Dynamic Programming: basic concept, use, examples (matrix-chain multiplication, all-pair shortest paths, single-source shortest path, travelling salesman problem).
Branch and Bound: basic concept, use, examples (15-puzzle problem).

Backtracking: basic concept, use, examples (Eight queens problem, graph coloring problem, Hamiltonian problem).
Greedy Method: basic concept, use, examples (Knapsack problem, Job sequencing with deadlines, minimum spanning tree).
Lower Bound Theory: Bounds on sorting and sorting techniques using partial and total orders.
Disjoint Set Manipulation: Set manipulation algorithm like UNION-FIND, union by rank, Path compression.

Module - 3 (12 hours)
Properties of graphs and graph traversal algorithms: BFS and DFS.

Matrix manipulation algorithms: Different types of algorithms and solution of simultaneous equations, DFT & FFT algorithm; integer multiplication schemes.

Notion of NP-completeness: P class, NP-hard class, NP-complete class, Circuit Satisfiability problem, Clique Decision Problem.
Approximation algorithms: Necessity of approximation scheme, performance guarantee, Polynomial time approximation schemes: 0/1 knapsack problem.

Suggested Books:

1. T.H. Cormen, C.E. Leiserson., R.L. Rivest, C. Stein: "Introduction to Algorithms",.
2. E. Horowitz, S. Sahni, S. Rajasekaran: "Fundamentals to Computer Algorithms",.
3. C.H. Papadimitriou, E. Steiglitz: "Combinatorial Optimization Algorithms and Complexity",.

DBMS Lab

Code: BCA292
Contacts: 3P
Credits: 2

Course Objectives:

- Students should be able to model, design and implement Data Base Management Systems
- Students should be able to manipulate a database using Structured Query Language (SQL)

Course Outcomes:

CO 1: Design and Implement a database schema
CO 2: Devise queries using DDL, DML, DCL and TCL commands.
CO 3: Applications using PL/SQL
CO 4: Design and implement a project using SQL and Programming Language.

CO – PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1		√			√		
CO2		√			√		
CO3	√	√	√		√		
CO4			√		√		

Overview of Structured Query Language

1. Creating Database

- Creating a Database
- Creating a Table
- Specifying Relational Data Types
- Specifying Constraints
- Creating Indexes

2. Table and Record Handling

- INSERT statement
- Using SELECT and INSERT together
- DELETE, UPDATE, TRUNCATE statements
- DROP, ALTER statements

3. Retrieving Data from a Database

- The SELECT statement
- Using the WHERE clause
- Using Logical Operators in the WHERE clause
- Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING

4. Clauses

- Using Aggregate Functions
- Combining Tables Using JOINS
- Subqueries

5. **Database Management**
   - Creating Views
   - Creating Column Aliases
   - Creating Database Users
   - Using GRANT and REVOKE

6. **Cursors in Oracle PL / SQL**

7. **Writing Oracle PL / SQL Stored Procedures**

Computer Graphics Lab

Code: BCA294
Contacts: 3P
Credits: 2

Course Objectives:

- To make students familiar about software, tools and interfaces available for computer graphics applications development.
- To make students familiar about different programming environments.
- To comprehend graphics algorithms through implementation.

Course Outcomes:

CO 1: Familiarity about software tools and interfaces available for computer graphics applications development.
CO 2: Knowledge about different programming environments and their applicability.
CO 3: Understanding various scan conversion and filling algorithms through implementation.
CO 4: Practical knowledge about 2D and 3D object representation, transformation and viewing.
CO 5: Having skillsets in aspects of graphics application development.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√	√	√		√
CO2	√	√	√	√	√		√
CO3	√	√	√	√	√		
CO4	√	√	√	√	√	√	√
CO5	√	√	√	√	√	√	√

Rasterization and plotting of points, lines, circles, ellipses, etc. and comparative analysis.
Various transformations of 2D objects – translation, rotation, shear, reflection, etc.
Filling algorithms – scanline, boundary fill, flood-fill; study of applicability.
Clipping of points, lines and 2D objects.
3D modeling, transformation and viewing.

Suggested Books and References:

1. S. Guha, "Computer Graphics Through OpenGL: From Theory to Experiments", CRC Press.
2. V. S. Gordon, J. L. Clevenger "Computer Graphics Programming in OpenGL with C++", Mercury Learning & Information publication.
3. Computer Graphics – A Programming Approach by S. Harrington, TMH publication.
4. G. Enderle, K. Kansy and G. Pfaff, "Computer Graphics Programming", Springer

Scripting Lab

Code: BCA296

Contacts: 3P

Credits: 2

Course Objectives:

- To provide an overview of the basic fundamentals of python language.
- To introduce types, operators and expressions.
- To familiarize with various types of data structures.
- To learn how to write functions and pass arguments in Python.
- To understand how to build and package Python modules for reusability.
- To develop object oriented skills in Python.

Course Outcomes:

CO 1: Students will be able to understand the necessity of python programming and be familiar with how to write python programs.
CO 2: Students will be able to familiar with various categories of types, operators and expressions.
CO 3: Students will be able to familiar with various types of data structures.
CO 4: Students will be able to learn how to write functions and pass arguments in Python.
CO 5: Students will be able to learn how to build and package Python modules for reusability.
CO 6: Students will be able to acquire object oriented skills in Python.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√	√	√		
CO2	√	√	√	√	√		
CO3	√	√	√	√	√		√
CO4	√	√	√	√	√	√	√
CO5	√	√	√	√	√	√	√
CO6	√	√	√	√	√	√	√

The lab experiments for this course have to ensure that the following concepts of PYTHON LANGUAGE are covered during lab classes:

Introduction: Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

Types, Operators and Expressions: Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations Control Flow- if, if-elif-else, for, while, break, continue, pass

Data Structures Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences. Comprehensions.

Functions - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions(Function Returning Values), Scope of the Variables in a Function - Global and Local Variables. **Modules:** Creating modules, import statement, from. Import statement, name spacing, **Python packages** Introduction to PIP, Installing Packages via PIP, Using Python Packages

Object Oriented Programming OOP in Python: Classes, 'self variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data hiding, **Error and Exceptions:** Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions

Suggested Books:

1. Vamsi Kurama: ―Python Programming: A Modern Approach‖, Pearson.
2. Mark Lutz: ―Learning Python‖, O,,Rielly
3. W.Chun: ―Core Python Programming‖, Pearson.
4. Introduction to Python, Kenneth A. Lambert, Cengage
5. S. Das: ―Unix System V.4 Concepts and Applications‖, 3rd Ed., Tata Mcgraw-Hill, 2013.
6. D. Flanagan: ―Javascript: The Definitive Guide‖, 5th Ed., O'Reilly, 2006.
7. D. Gosselin: ―PHP Programming with MySQL‖, Course Technology, 2006.

Semester V:

Software Engineering

Code: BCA301
Contracts: 3L
Credits: 3

Course Objectives:

- To acquire the generic software development skill through various stages of software life cycle.
- To ensure the quality of software through software development with various protocol based environment.

Course Outcomes:

CO 1: Basic knowledge and understanding of the analysis and design of complex systems.
CO 2: Ability to apply software engineering principles and techniques.
CO 3: Ability to develop, maintain and evaluate large-scale software systems.

CO 4: To produce efficient, reliable, robust and cost-effective software solutions.

CO 5: Ability to perform independent research and analysis.

CO 6: Ability to work as an effective member or leader of software engineering teams.

CO 7: Ability to understand and meet ethical standards and legal responsibilities

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2		√	√				
CO3	√		√				
CO4			√	√		√	
CO5		√	√				√
CO6					√	√	
CO7							√

Module 1: **[12 Hrs]**

Overview: System Analysis & Design, Business System Concept, System Development Life Cycle, Waterfall Model, Spiral Model, Prototype Model, Feasibility Analysis, Technical Feasibility, Cost-Benefit Analysis, COCOMO model.

System Requirement Specification: DFD, Data Dictionary, ER diagram,

System Design: Problem Partitioning, Top-Down and Bottom-Up design; Decision tree, decision table and structured English; Functional vs. Object- Oriented approach.

Module 2: **[12 Hrs]**

Coding & Documentation: Structured Programming, OO Programming, Information Hiding, Reuse, System Documentation.

Testing: Levels of Testing, Integration Testing, Test case Specification, Reliability Assessment, Validation & Verification Metrics, Monitoring & Control.

Module 3: **[12 Hrs]**

Software Project Management: Project Scheduling, Staffing, Software Configuration Management, Quality Assurance, Project Monitoring.

CASE tools: Concepts, use and application.

UML diagrams: Fundamentals of Object Oriented design in UML.

References:

1. Roger Pressman: “Software Engineering, A practitioner‟s approach”,

2. Rajib Mall: “Software Engineering”,

2. Pankaj Jalote: “Software Engineering”,

Microprocessor

Code: BCA303

Contracts: 3L

Credits: 3

Course Objectives:

- To learn the basics of a particular microprocessor.
- To learn the basics of a particular microcontroller.
- To learn the interfacing of microprocessor.

Course Outcomes:

CO 1: To acquire the knowledge of hardware details of 8085 and 8086 microprocessor with the related signals and their implications.

CO 2: To develop skill in assembly Language programming of 8085.

CO 3: To understand the concept and techniques of designing and implementing interfacing of microprocessor with memory and peripheral chips involving system design.

CO 4: To acquire the knowledge of the 8086 architecture and its programming.

CO 5: To analyze the performance of computers and its architecture to real-life applications

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√		√				
CO2	√		√	√			√
CO3	√		√	√		√	
CO4	√		√				√
CO5	√			√		√	

Module I: [6 Hrs]

Introduction to Microprocessors and their features. Demultiplexing of Address & Data Bus, Generation of Read Write Control Signal for Memory & I/O. Uses of Decoder & Latch, Changes of Memory Map.

Module II: [9 Hrs]

8085 Microprocessor: Architecture, Register Organization, Control Signals, Hardware & Software Interrupts, Instruction Set, Addressing Modes & Assembly Language Programming.

Instruction Cycle, Machine Cycle, T-State, Timing Diagram [**03**L]

Interfacing of Memory Chips with Microprocessor, I/O Mapped I/O & Memory Mapped I/O. [**06L**]

Module III: [14 Hrs]

Working of DMA Controller, Microprocessor based A/D Conversion using ADC0804. [**03**L]
Peripherals: 8255, 8253, 8259, 8237 [**09**L]
Industrial Applications of Microprocessor. [**02**L]

Module IV: [11 Hrs]

8086 Microprocessor: Architecture, Memory Segmentation, Minimum Mode & Maxamimum Mode,

Addressing Modes, brief description of Instruction Set and Assembly Language Programming. [9**L**]
Brief overview of some other Microprocessors [**02L**]

Suggested Books:

1. Ramesh Gaonkar: "Microprocessor Architecture, Programming, and Applications with the 8085", Penram International Publishing (India) Private Limited.
2. B. Ram: "Fundamentals of Microprocessors and Microcomputers", Dhanpat Rai Publications.
3. N. Senthil Kumar, M. Saravanan and S. Jeevananthan: "Microprocessors and Microcontrollers", Oxford University Press.
4. A.K. Ray and K M Bhurchandi: "Advanced Microprocessors and Peripherals: Architecture, Programming and Interfacing", Tata McGraw-Hill Publishing Company Limited.
5. M. A Mazidi, J .G. Mazidi and R. D. McKinlay: "The 8051 Microcontroller and Embedded Systems:Using Assembly and C", Pearson.
6. Kenneth Ayala: "The 8051 Microcontroller", Cenage Learning India Private Limited.
7. http://nptel.ac.in/.

Web Technology

Code: BCA305
Contracts: 3L
Credits: 3

Course Objectives:

- To impart the basics of web page design.
- To understand important components of HTML5 documents and use HTML5 to create web pages.
- To learn to use JavaScript in Webpages to enhance the functionality and appearance of web pages.
- To know XML schema and transformation.
- To design dynamic web pages using PHP.

Course Outcomes:

CO 1: Student will be able to summarize the basic tags and properties in HTML and CSS.
CO 2: Student will be able to select HTML tags and CSS properties to design web pages.
CO 3: Student will be able to prepare XML documents to store and transport data.
CO 4: Student will be able to write programs in PHP.
CO 5: Student will be able to develop web applications using JavaScript and PHP.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√	√				
CO2		√	√		√		
CO3			√	√			
CO4			√		√		
CO5		√	√		√		

Static Web Pages [5 Hrs]
HTML: Introduction, Editors, Elements, Attributes, Heading, Paragraph. Formatting, Link, Head, Table, List, Block, Layout,

Dynamic Web Pages [4 Hrs]

The need of dynamic web pages; an overview of DHTML, Cascading Style Sheet (CSS), comparative studies of different technologies of dynamic page creation

Active Web Pages [5 Hrs]
Need of active web pages; Java Applets: Container Class, Components, Applet Life Cycle, Update method; Parameter passing Applet, Applications.

Java Script [6 Hrs]
Data types, variables, operators, conditional statements, Array object, Date object, String object, Function, Errors, Validation.

Extensible Markup Language (XML) [5 Hrs]
Introduction, Tree, Syntax, Elements, Attributes, Validation, Viewing. XHTML in brief.

Cookies & Sessions [4 Hrs]
Definition of cookies; Create and Store a cookie with example; Sessions.

Java Servlet [5 Hrs]
Servlet environment and role, HTML support, Servlet API, The Servlet life cycle, Servlet Programs.

JSP [15 Hrs]
JSP architecture, JSP servers, JSP tags, understanding the layout in JSP, Declaring variables, methods in JSP, inserting java expression in JSP, processing request from user and generating dynamic response for the user, inserting applets and java beans into JSP, using include and forward action, comparing JSP and CGI program, comparing JSP and ASP program; Creating ODBC data source name, introduction to JDBC, prepared statement and callable statement.

PHP & MySQL [5 Hrs]
Overview of PHP, Basics web programming using PHP, Introducing MySQL, Database connectivity using PHP.

Suggested Books:

1. Uttam K. Roy: “Web Technologies”, Oxford University Press.
2. Ivan Bayross, Sharanam Shah, Cynthia Bayross, Vaishali Shah: “Java Server Programming for Professionals”, Shroff Publishers and Distributors.
3. C. Xavier: “Web Technology and Design”, New Age.
4. Kogent Learning Solutions Inc.: “Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML and Ajax, Black Book: HTML, Javascript, PHP, Java, Jsp, XML and Ajax, Black Book”, Dreamtech Press.
5. N.P. Gopalan and J. Akilandeswari: “Web Technology: A Developer's Perspective”, PHI.

6. Luke Welling, Laura Thomson: "PHP and MySQL Web Development", Pearson Education.
7. Mike McGrath: "PHP and MySQL", McGraw Hill Education.
8. Meloni J C: "Sams Teach Yourself PHP MY SQL and Apache", Pearson Education.
9. Jain & Siddiqui with NIIT: "J2EE Professional Projects", PHI.
10. Uttam K. Roy: "Advanced Java Programming", Oxford University Press.

Elective-I

Code: BCA307
Contracts: 3L
Credits: 3

Environmental Studies

Code: ES331
Contracts: 3L
Credits: 3

Microprocessor Lab

Code: BCA393
Contracts: 3P
Credits: 2

Course Objectives:

- To familiar with 8085 microprocessor kit and its functioning.
- Able to run assembly language programming.
- Learn to interface peripherals with of microprocessor.
- To be familiar with application microprocessor and microcontroller in real life.

Course Outcomes:

CO 1: Knowledge about the fundamentals 8085 trainer KIT.
CO 2: Capable to write assembly level programming.
CO 3: Know how to troubleshoot interactions between software and hardware.
CO 4: Know how to interface peripheral devices with microprocessor.
CO 5: Able to realize the memory mapping in microprocessor.

CO – PO Mapping:

	PSO1	PSO2	PSOP3	PSO4	PSO5	PSO6	PSO7
CO1	√				√		

CO2	√				√		√
CO3	√	√	√	√	√		
CO4	√	√	√	√	√		
CO5	√				√		√

Familiarization with 8085 register level architecture and trainer kit components, including the memory map. Familiarization with the process of storing and viewing the contents of memory as well as registers.

Study of prewritten programs on trainer kit using the basic instruction set (data transfer, Load/Store, Arithmetic, Logical). Assignments based on above.

Familiarization with 8085 simulator on PC. Study of prewritten programs using basic instruction set (data transfer, Load/Store, Arithmetic, Logical) on the simulator. Assignments based on above.

Programming using kit/simulator for Table look up

Copying a block of memory Shifting a block of memory

Packing and unpacking of BCD numbers Addition of BCD numbers

Binary to ASCII conversion String Matching

Multiplication using Booth‟s Algorithm

Program using subroutine calls and IN/OUT instructions using 8255 PPI on the trainer kit eg, subroutine for delay, reading switch state & glowing LEDs accordingly, finding out the frequency of a pulse train etc.

Interfacing any 8-bit Latch (e.g., 74LS373) with trainer kit as a peripheral mapped output port with absolute address decoding.

Interfacing with I/O modules: ADC, Speed control of mini DC motor using DAC, Keyboard, Multi-digit Display with multiplexing, Stepper motor.

Web Technology Lab

Code: BCA395

Contracts: 3P

Credits: 2

Course Objectives:

- To acquire knowledge and skills for creation of website considering both client- and server-side Programming.
- To create Web application using tools and techniques used in industry.
- To Demonstrate Amazon/Google or Yahoo mash up
- To be well versed with XML and web services Technologies.
- To be familiarized with open source Frameworks for web development.

Course Outcomes:

CO 1: Creating web page using the knowledge of HTML and CSS.

CO 2: Implement dynamic web pages with validation using JavaScript objects by applying different event handling mechanism.

CO 3: Develop simple web application using server side PHP programing and Database Connectivity using MySQL.

CO 4: Demonstrate simple web application using Python Django Framework.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√	√		√		
CO2	√	√	√		√		
CO3	√		√	√			
CO4		√	√	√	√		

1. Web Page Design using HTML
2. Use of CSS in Designing Web Pages
3. Applet Design
4. Application of JavaScript in Web Page Development
5. Usage of Cookies & XML
6. Server Side Programming through Servlets
7. Application of Java Server Pages in Server Side programming
8. Application of Java Database Connectivity
9. Web design using PHP and MySQL

Suggested Books:

1. Uttam K. Roy: "Web Technologies", Oxford University Press.
2. Ivan Bayross, Sharanam Shah, Cynthia Bayross, Vaishali Shah: "Java Server Programming for Professionals", Shroff Publishers and Distributors.
3. C. Xavier: "Web Technology and Design", New Age.
4. Kogent Learning Solutions Inc.: "Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML and Ajax, Black Book: HTML, Javascript, PHP, Java, Jsp, XML and Ajax, Black Book", Dreamtech Press.

Software Engineering Lab

Code: BCA397

Contracts: 3P

Credits: 2

Course Objectives:

- To acquire the generic software development skill through various stages of software life cycle.
- To ensure the quality of software through software development with various protocol based environment.

Course Outcomes:

CO 1: Basic knowledge and understanding of the analysis and design of complex systems.
CO 2: Ability to apply software engineering principles and techniques.
CO 3: Ability to develop, maintain and evaluate large-scale software systems.
CO 4: To produce efficient, reliable, robust and cost-effective software solutions.
CO 5: Ability to perform independent research and analysis.
CO 6: Ability to work as an effective member or leader of software engineering teams.
CO 7: Ability to understand and meet ethical standards and legal responsibilities.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2		√	√				
CO3	√		√				
CO4			√	√		√	
CO5		√	√				√
CO6					√	√	
CO7							√

1. Preparation of requirement document for standard application problems in standard format.(e.g Library Management System, Railway Reservation system, Hospital management System, University Admission system)
2. Project Schedule preparation .
3. Use Case diagram,Class diagram,Sequence diagram and prepare Software Design Document using tools like Rational Rose.(For standard application problems)
4. Estimation of project size using Function Point(FP) for calculation.
5. Design Test Script/Test Plan(both Black box and White Box approach)
6. Compute Process and Product Metrics (e.g Defect Density,Defect Age,Productivity,Cost etc.) also by Cost Estimation

SEMESTER-VI

Elective-II

Code: BCA302
Contacts: 3L
Credits: 3

Introduction to Data Science

Code: BCA304
Contacts: 3L
Credits: 3

Course Objectives:

- To enable the student to understand basic data science concepts.
- To learn to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization.
- To make aware of how to address advanced statistical situations
- To understand regression and clustering.

- To understand feature generation and feature selection

Course Outcomes:

CO 1: Students should be able to understand the problem.
CO 2: Able to comprehend the problem.
CO 3: Able to define suitable statistical model to be adopted.
CO 4: Able to apply different Data Analysis Techniques.
CO 5: Able to generate features and select features from meaningful data.
CO 6: Understand different case studies.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√		√			√
CO2		√		√		√	√
CO3	√	√		√			
CO4		√			√	√	
CO5.		√	√	√			
CO6	√	√		√	√		√

1. Introduction: What is Data Science? - Big Data and Data Science hype – and getting past the hype - Why now? – Datafication - Current landscape of perspectives - Skill sets needed
2. Statistical Inference - Populations and samples - Statistical modeling, probability distributions, fitting a model - Intro to R
3. Exploratory Data Analysis and the Data Science Process - Basic tools (plots, graphs and summary statistics) of EDA - Philosophy of EDA - The Data Science Process - Case Study: RealDirect (online real estate firm)
4. Three Basic Machine Learning Algorithms - Linear Regression - k-Nearest Neighbors (k-NN) - k-means
5. Linear Regression and k-NN, Naive Bayes classifier
6. Feature Generation and Feature Selection (Extracting Meaning From Data) - Motivating application: user (customer) retention - Feature Generation (brainstorming, role of domain expertise, and place for imagination) - Feature Selection algorithms – Filters; Wrappers; Decision Trees; Random Forests

Suggested Books

1. Cathy O‟Neil and Rachel Schutt. Doing Data Science, Straight Talk From The Frontline. O‟Reilly. 2014.
2. Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. Mining of Massive Datasets., Cambridge University Press. 2014. (free online)
3. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. ISBN 0262018020. 2013.
4. Foster Provost and Tom Fawcett. Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking. ISBN 1449361323. 2013

Data science Lab:

Code: BCA392

Contacts: 3L
Credits: 3

Course Objectives:

- To enable the student to understand basic data science concepts.
- To learn to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization.
- To make aware of how to address advanced statistical situations
- To understand regression and clustering.
- To understand feature generation and feature selection

Course Outcomes:

CO 1: Students should be able to understand the problem.
CO 2: Able to comprehend the problem in R or Python Language.
CO 3: Able to define suitable statistical model to be adopted.
CO 4: Able to apply different Data Analysis Techniques.
CO 5: Able to generate features and select features from meaningful data.
CO 6: Understand different case studies.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√		√			√
CO2		√		√		√	√
CO3	√	√		√			
CO4		√			√	√	
CO5.		√	√	√			
CO6	√	√		√	√		√

1. Introduction to R and Python language.
2. Data collection - data wrangling, cleaning, and sampling to get a suitable data set;
3. Data management - accessing data quickly and reliably;
4. exploratory data analysis – generating hypotheses and building intuition, prediction or statistical learning; and communication – summarizing results through visualization, stories, and interpretable summaries.
5. Visualization of data, performing clustering, classification using different packages of R.

Suggested Books:

1. Martelli, Alex, Anna Ravenscroft, and David Ascher. *Python cookbook*. " O'Reilly Media, Inc.", 2005.
2. Beazley, David M. *Python essential reference*. Addison-Wesley Professional, 2009.
3. Matloff, Norman. *The art of R programming: A tour of statistical software design*. No Starch Press, 2011.

Seminar and Presentation Skills

Code: BCA396
Contacts: 3P
Credits: 2

Elective-I & II

A. Big Data Analytics

Course Objectives:

- To enable the student to understand basic Big data analytics concepts.
- To learn to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization.
- To make aware of how to address advanced statistical situations
- To understand regression and clustering.
- To understand feature generation and feature selection

Course Outcomes:

CO 1: Students should be able to understand the problem.
CO 2: Able to comprehend the problem.
CO 3: Able to define suitable statistical model to be adopted.
CO 4: Able to apply different Big Data Analytics Techniques.
CO 5: Able to generate features and select features from meaningful data.
CO 6: Understand to apply unsupervised learning.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√		√			√
CO2		√		√		√	√
CO3	√	√		√			
CO4		√			√	√	
CO5		√	√	√			
CO6	√	√			√		√

Module-1 (12 hours)

Introduction, Big Data Overview, What is data sciences, The rising and importance of data sciences, Big data analytics in industry verticals, Data Analytics Lifecycle and methodology , Data Understanding Data Preparation, Modeling, Evaluation, Communicating results, Deployment, Data exploration & preprocessing , Measures and evaluation.

Module-2 (12 hours)

Data Analytics: Theory & Methods, Supervised learning, Decision trees, Naïve Bayes, SVM, Linear Regression: gradient descent, Normal equations. Probabilistic Interpretation, Logistic Regression, Newton's method, Locally weighted Linear Regression, Exponential Families, Generalized Linear Models, Optimization: Convex functions, Convex problems, Generative Learning Algorithms, Gaussian Discriminant Analysis, Feature selection, Kernels, Support vector Machine.

Module-3 (12 hours)

Unsupervised Learning: Curse of Dimensionality, Dimensionality Reduction, PCA, Mixture of Gaussians, EM Algorithm. Examples of EM, clustering, spectral clustering. Multi-dimensional Scaling (MDS), Isomaps, Non Negative Matrix Factorization.

Reference books

1. Gaussian Processes for Machine Learning Carl Edward Rasmussen and Christopher K.I. Williams, MIT Press, 2006.
2. Non-Uniform Random Variate Generation Luc Devroye, Springer-Verlag, 1986.
3. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, EMC Education Services (Editor).

B. Soft Computing

Course Objectives:

- To familiarize with soft computing concepts.
- To introduce the fuzzy logic concepts, fuzzy principles and relations.
- To Basics of ANN and Learning Algorithms.
- Ann as function approximation.
- Genetic Algorithm and its applications to soft computing. 6. Hybrid system usage, application and optimization.

Course Outcomes:

CO 1: List the facts and outline the different process carried out in fuzzy logic, ANN and Genetic Algorithms.
CO 2: Explain the concepts and meta-cognitive of soft computing.
CO 3: Apply Soft computing techniques to solve character recognition, pattern classification, regression and similar problems.
CO 4: Outline facts to identify process/procedures to handle real world problems using soft computing.
CO 5: Evaluate various techniques of soft computing to defend the best working solutions.
CO 6: Design hybrid system to revise the principles of soft computing in various applications.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2	√						
CO3	√	√			√		
CO4				√	√	√	
CO5			√		√	√	
CO6				√	√		

Module-1 (12 hours)

Introduction: Soft Computing, Components of Soft Computing, Importance of Soft Computing, Applications. Fuzzy Set Theory:
Definition, Different types of fuzzy set membership functions. Fuzzy set theoretic operations, Fuzzy

rules and fuzzy reasoning, Fuzzy inference systems.

Module-2 (12 hours)

Introduction to Optimization problem. Gradiant descent algorithm, Genetic Algorithms, Simulated Annealing.

Neural Networks: Artificial neural networks models,

Clustering and its Applications: K-means, K-medoids, Hierarchical Clustering.

Module-3 (12 hours)

Classification and Predictions: What is Classification & Prediction, Issues regarding Classification and prediction, Decision tree,

Bayesian Classification, Classification by Back propagation, Multilayer feed-forward Neural Network,

Dimensionality Reduction

Introduction Principal Components Analysis, Singular Value Decomposition, Multidimensional Scaling

Suggested Books:

1. J. S. R. Jang, C.T.Sun and E.Mizutan: "Neuro Fuzzy and Soft Computing", PHI.
2. Michalski: "Machine Learning - An A. I. Approach", Carbonnel & Michel (Eds.)
3. J. Ross Timothy: "Fuzzy Logic with Engineering Applications", John Wiley and Sons.
4. Rajasekaran and G. A. V. Pai: "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI, 2003.

C. Machine Learning Fundamentals

Course Objectives:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To explain mathematical background of the Machine Learning Algorithms
- To develop skills for solving practical problems using Machine Learning

Course Outcomes:

Students will be able to:

CO 1: Recognize characteristics of machine learning that make it useful in certain types of real-world analysis problems.
CO 2: Understanding machine learning problems as supervised, semi-supervised, and unsupervised.
CO 3: Become familiar with feature engineering and Dimensionality reduction Techniques
CO 4: implementing regression, clustering, classification, and reinforcement task
CO 5: Using Python Machine Learning libraries

CO – PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	√	√					
CO2		√		√			
CO3	√	√		√			
CO4		√		√	√		
CO5				√	√		

Module 1 (18 hours)

Introduction: Overview of topics and applications
Supervised Learning: Linear Regression: gradient descent, Normal equations. Probabilistic Interpretation, Logistic Regression, Newton's method, Locally weighted Linear Regression, Nearest Neighbors, Exponential Families, Generalized Linear Models, Optimization: Convex functions, Convex problems, Generative Learning Algorithms, Gaussian Discriminant Analysis, Feature selection, Kernels, Support vector Machine.

Module 2 (18 hours)

*Unsupervised Learning:*Curse of Dimensionality, Dimensionality Reduction, PCA, Mixture of Gaussians, EM Algorithm. Examples of EM, clustering, spectral clustering. Multi-dimensional Scaling (MDS), Isomaps, Non Negative Matrix Factorization.

Module 3 (12 hours)

Probabilistic Graphical Models: Introduction, Representation, Markov Blanket, variable elimination, HMM, Inference on a chain (sum-product specific case), Kalman Filters, Directed / Undirected graphs, MRFs, Sum-product, Max-product,
Special Topic: Graphical Models, Deep Learning.

Reference Books:

- Machine Learning, Tom Mitchell, McGraw Hill
- The Elements of Statistical Learning Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer
- EthemAlpaydin, Introduction to Machine Learning, PHI
- Chris Bishop, Pattern Recognition and Machine Learning

D. Digital Signal Processing

Course Objectives:

- The primary objective of this course is to provide a thorough understanding and working knowledge of design, implementation and analysis DSP systems.

Course Outcomes:

CO 1: Interpret, represent and process discrete/digital signals and systems thorough understanding of frequency domain analysis of discrete time signals.
CO 2: Ability to design & analyze DSP systems like FIR and IIR Filter etc.
CO 3: Practical implementation issues such as computational complexity, hardware resource.
CO 4: Limitations as well as cost of DSP systems or DSP Processors.
CO 5: Ability to perform independent research and analysis.Understanding of spectral analysis of the signals.

CO – PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	√	√					
CO2		√	√				
CO3	√		√				
CO4			√	√		√	

CO5		√	√		√		√

Module-1 (12 hours):

Discrete-time signals: Discrete-time sequences, their frequency domain behaviour, comparison with analog signals, convolution of two sequences, sampling theorem, Reconstruction of continuous- time signals. Unit-sample response of a system, Time-invariant systems, Superposition principle for linear systems, Stability criterion for discrete-time systems, Causality criterion for discrete-time systems, Linear constant-coefficient difference equations.

Module-2 (12 hours):

Discrete-time Fourier transform: FT of special sequences, the inverse FT; Computation of the DFT from the discrete-time sequence, linear and circular convolution; computations for evaluating the DFT: increasing the computational speed of the DFT.
Z-transform: Definition and properties of the z-transform, the inverse z-transform; relationship between the Fourier transform and the z-transform.

Module-3(12 hours):

Digital filter: filter categories: IIR and FIR, recursive and non-recursive. Digital Filter Structures: The direct form I and II structures, Cascade combination of second-order sections, parallel combination of second-order sections, Linear-phase FIR filter structures, Polyphase decomposition;
Frequency-sampling structure for the FIR filter. Uniform DFT filter banks.
Digital Signal Processor: Architecture of TMS320C 5416/6713 Processor (any one; programs in Assembly Language.

Suggested Readings:

1. Digital Signal Processing – Principles, Algorithms and Applications - J.G.Proakis & D.G. Manolakis
2. Digital Signal Processing- Alan V. Oppenheim, Ronald W. Schafer
3. Digital Signal Processors Architectures, Implementations and Applications – S.M.Kuo & W. Gan

E. Pattern Recognition

Course Objectives:

- To understand concepts and basic approaches to development of pattern recognition and machine intelligence systems.
- To learn the basic methods of feature extraction, feature evaluation, and classification.
- To understand and apply supervised and unsupervised classification methods to detect and characterize patterns in real-world data.

Course Outcomes:

CO 1: Familiarity to pattern recognition concepts, issues and approaches.

CO 2: Ability to explain and compare a variety of pattern classification methods.
CO 3: Ability to analyze different clustering and classification problems and solve using pattern recognition techniques.
CO 4: Knowledge about evaluation methods for supervised and unsupervised pattern analysis techniques.
CO 5: Application of pattern recognition techniques in real-world problems.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√	√	√	√	
CO2	√	√	√	√	√		
CO3	√	√	√	√	√		
CO4	√	√	√	√	√		
CO5	√	√	√	√	√	√	√

Module -1 (12 hours)

Introduction and Mathematical Preliminaries: What Is Pattern Recognition? Clustering vs. Classification; Applications; Linear Algebra, Vector Spaces, Probability Theory, Estimation Techniques.

Classification: Bayes Decision Rule, Error Probability, Error Rate, Minimum Distance Classifier, Mahalanobis Distance; K-NN Classifier, Linear Discriminant Functions and Non-Linear Decision Boundaries. Fisher‟s LDA, Single and Multilayer Perceptron, Training Set and Test Sets, Standardization and Normalization.

Module -2 (12 hours)

Clustering: Different Distance Functions and Similarity Measures, Minimum Within Cluster Distance Criterion, K-Means Clustering, Single Linkage and Complete Linkage Clustering, K- Medoids, DBSCAN, Visualization of Datasets, Existence Of Unique Clusters or No Clusters.

Module -3 (12 hours)

Feature Selection: Problem Statement And Uses, Probabilistic Separability Based Criterion Functions, Interclass Distance Based Criterion Functions, Branch And Bound Algorithm, Sequential Forward/Backward Selection Algorithms, (l,r) Algorithm.

Feature Extraction: PCA, Kernel PCA.

Suggested Books:

1. K. Fukunaga, Statistical pattern Recognition, Academic Press, 2000.
2. S.Theodoridis and K.Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009.
3. V.S. Devi, M.N. Murty, Pattern Recognition: An Introduction, Universities Press, Hyderabad.
4. R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, Wiley, 2000.

F. Bioinformatics

Course Objectives:

- Knowledge learned without implementation (content covered in the lecture component of contact sessions).
- The basic objective is to give students an introduction to the basic practical techniques of bioinformatics. Emphasis will be given to the application of bioinformatics and biological databases to problem solving in real research problems.

Course Outcomes:

CO 1: To get introduced to the basic concepts of Bioinformatics and its significance in Biological data analysis.
CO 2: Describe the history, scope and importance of Bioinformatics.
CO 3: Explain about the methods to characterize and manage the different types of Biological data.
CO 4: Classify different types of Biological Databases.
CO 5: Introduction to the basics of sequence alignment and analysis.
CO 6: Overview about biological macromolecular structures and structure prediction methods.

CO –PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2		√	√				
CO3	√		√				
CO4			√	√			
CO5		√	√				√
CO6					√	√	

Module-1 (12 hours)

Background: Why computational biology, biological information, challenges in computational biology.

Sequence Assembly: Fragment assembly, Sequencing by hybridization, Overlap-layout-consensus

Sequence Alignment: Introduction to biological sequences, DNA sequence, dynamic programming methods for global and local alignment, gap penalty functions, heuristics in alignment, BLAST, pairwise sequence alignment, multiple sequence alignment

Module-2 (12 hours)

Phylogenetic Trees: Distance, parsimony, and probabilistic methods of phylogenetic tree construction, models of sequence evolution

Annotating genomes: Markov chains, high-order Markov models, Forward/Backward/Viterbi algorithms, applications to genome segmentation and annotation.

Module-3 (12 hours)

Clustering approaches to biological datasets: High-throughput technologies, clustering algorithms, evaluation of clusters

Analysis of gene expression data

Modeling and analysis of biological networks: Biological networks, computational problems in network biology, Bayesian networks, module networks, parameter and structure learning, regression-based network inference, network applications.

Machine Learning algorithms and its usage in modeling biological data

Suggested Books:

1. JIN XIONG: Essential Bioinformatics, Cambridge University Press
2. Joachim Selbig and Stefanie Hartmann: Introductory Bioinformatics: Fourth Edition
3. Jonathan Pevsner: Bioinformatics and Functional Genomics, Wiley- Blackwell
4. S.C. Rastogi, N Mendiratta, P Rastogi: Bioinformatics: Methods & Applications, PHI
5. Stanley I. Letovsky: Bioinformatics: Databases and Systems.
6. Sorin Draghici: Bioinformatics Databases: Design, Implementation, and Usage (Chapman & Hall/ CRC Mathematical Biology & Medicine).
7. Arthur M.Lesk: Data base annotation in molecular biology, Principles and Practices.
8. Tao, Jiang, Ying Xu, Michael Q. Zang: Current topics in computational molecular biology.

G. Natural Language Processing

Course Objectives:

- Student should understand the theoretical concepts of natural language processing in Linguistics and Formal Language theory
- Enable students to be capable to syntactic, semantic and pragmatic processing of Natural Languages
- Student should be able to analyze NLP models and algorithms using both the traditional and the statistical approaches

Course Outcome:

CO 1: Understanding fundamentals of syntax, semantics and rules in NLP and Automata Theory
CO 2: Basic Text Processing Operations
CO 3: Learning Language modeling techniques
CO 4: Understanding Information Retrieval Techniques
CO 5: Performing Text Classification using Machine Learning

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√	√				
CO2			√	√			
CO3	√	√	√	√			
CO4	√	√		√	√		
CO5	√		√				

Unit I. Regular Expressions and Automata (4L)

Introduction to NLP, Regular Expression, Finite State Automata

Unit II. Basic Text Processing (8L)

Word Tokenization, Normalization, Sentence Segmentation, Named Entity Recognition, Multi Word Extraction, Spell Checking – Bayesian Approach, Minimum Edit Distance; Morphology – Inflectional and Derivational Morphology, Finite State Morphological Parsing, The Lexicon and Morphotactics, Morphological Parsing with Finite State Transducers, Orthographic Rules and Finite State Transducers, Porter Stemmer

Unit III. Language Modeling (10L)

Introduction to N-grams, Chain Rule, Smoothing – Add-One Smoothing, Witten-Bell Discounting; Backoff, Deleted Interpolation, N-grams for Spelling and Word Prediction, Evaluation of language models; Markov Chain, Hidden Markov Models, Forward Algorithm, Viterbi Algorithm, Part of Speech Tagging – Rule based and Machine Learning based approaches, Evaluation

Unit IV. Text Classification (8L)

Text Classification, Naïve Bayes‟ Text Classification, Evaluation, Sentiment Analysis –Opinion Mining and Emotion Analysis, Resources and Techniques

Unit V. CFG and Lexical Semantics (8L)

Context Free Grammar and Constituency, Some common CFG phenomena for English, Top-Down and Bottom-up parsing, Probabilistic Context Free Grammar, Dependency Parsing; Introduction to Lexical Semantics – Homonymy, Polysemy, Synonymy, Thesaurus –WordNet, Computational Lexical Semantics – Thesaurus based and Distributional Word Similarity

Unit VI. Information Retrieval (10L)

Boolean Retrieval, Term-document incidence, The Inverted Index, Query Optimization, Phrase Queries, Ranked Retrieval – Term Frequency – Inverse Document Frequency based ranking, Zone Indexing, Query term proximity, Cosine ranking, Combining different features for ranking, Search Engine Evaluation, Relevance Feedback

Suggested Books:

1. Jurafsky and Martin: Speech and Language Processing, Pearson Education.
2. Manning and Schutze: Foundation of Statistical Natural Language Processing, MIT.

H. Visual Programming &Multimedia

Course Objectives:

- Explore the IDE environment and learn about the variables, data types and syntax rules used in program development.
- Learn to apply the decision structures, loop structures and functions for creating applications.
- Explore the details of multimedia system architecture and the multimedia databases.
- Ability to understand the multimedia family of standard like JPEG and MPEG.
- Learn about the various storage management systems and retrieval technologies.
- Learn about the different compression techniques of audio, images and video.

Course Outcomes:

CO 1: Have knowledge about the foundations of Visual Programming and Multimedia applications.

CO 2: Apply different rules and syntax of Visual Programming and write decision structures and loop structures for creating manageable code.
CO 3: Study the representations of different multimedia data and data formats.
CO 4: Comprehend the ideas of different compression techniques.
CO 5: Understand the different formats of audio and video files.
CO 6: Comprehend the concept of different coding techniques in multimedia applications and apply them to solve real world problems.
CO 7: Understand the comprehension of various storage media available for building various multimedia applications.
CO 8: Develop and evaluate various multimedia applications.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√					
CO2	√	√	√				
CO3	√	√	√				
CO4	√	√	√				
CO5		√	√				
CO6			√	√			
CO7			√	√			
CO 8		√		√	√		

Module 1: (12 Hrs)

WINDOWS PROGRAMMING

Windows environment – a simple windows program – windows and messages – creating the window – displaying the window – message loop – the window procedure – message processing – text output – painting and repainting – introduction to GDI – device context – basic drawing – child window controls

VISUAL C++ PROGRAMMING – INTRODUCTION

Application Framework – MFC library – Visual C++ Components – Event Handling – Mapping modes – colors – fonts – modal and modeless dialog – windows common controls – bitmaps

THE DOCUMENT AND VIEW ARCHITECTURE

Menus – Keyboard accelerators – rich edit control – toolbars – status bars – reusable frame window base class – separating document from its view – reading and writing SDI and MDI documents – splitter window and multiple views .

Module 2: (12 Hrs)

ACTIVEX AND OBJECT LINKING AND EMBEDDING (OLE)

ActiveX controls Vs. Ordinary Windows Controls – Installing ActiveX controls – Calendar Control – ActiveX control container programming – create ActiveX control at runtime.

Database Management with Microsoft ODBC – Structured Query Language – MFC ODBC classes – sample database applications – filter and sort strings – DAO concepts – displaying database records in scrolling view – Threading

Module 3: (12 Hrs)

Multimedia

Introduction

Multimedia today, Impact of Multimedia, Multimedia Systems, Components and Its Applications

Text and Audio

Text: Types of Text, Ways to Present Text, Aspects of Text Design, Character, Character Set, Codes, Unicode, Encryption; Audio: Basic Sound Concepts, Types of Sound, Digitizing Sound, Computer Representation of Sound (Sampling Rate, Sampling Size, Quantization),

Image and Video

Image: Formats, Image Color Scheme, Image Enhancement; Video: Analogue and Digital Video, Recording Formats and Standards (JPEG, MPEG, H.261) Transmission of Video Signals,

Module 4: (10 Hrs)

Storage models and Access Techniques

Magnetic media, optical media, file systems (traditional, multimedia) Multimedia devices – Output devices, CD-ROM, DVD, Scanner, and CCD

Image and Video Database

Image representation, segmentation, similarity based retrieval, image retrieval by color, shape and texture; indexing- k-d trees, R-trees, quad trees; Case studies- QBIC, Virage. Video Content,

Suggested Book:

1. Charles Petzold, "Windows Programming", Microsoft Press, 1998
2. David J. Kruglinski, George Shepherd, Scot Wingo, "Programming Microsoft
3. Visual C++", Microsoft Press, 2006
4. Kate Gregory „Using Visual C++", Prentice Hall of India Pvt., Ltd., 1999.
5. Buford J. K. – "Multimedia Systems" – Pearson Education.
6. Andleigh & Thakrar, "Multimedia", PHI
7. Balagurusamy E, "Programming in C# ", Tata McGraw Hill, 2010

I. Modeling and Simulation

Course Objectives:

- Define the basics of simulation modeling and replicating the practical situations in organizations
- Introduce various system modeling and simulation techniques
- describe the components of continuous and discrete systems and simulate them
- Understand different methods for random number generation
- Solve real world problems which cannot be solved strictly by mathematical approaches
- Develop simulation model using heuristic methods.
- Analysis of Simulation models using input analyzer, and output analyzer
- Explain Verification and Validation of simulation model

Course Outcomes:

CO 1: Understand the role of important elements of discrete event simulation and modeling paradigm.
CO 2: Develop a clear understanding of the need for the development process.
CO 3: Be able to explain the components of continuous and discrete systems and simulate them.
CO 4: Discuss the simulation methods and select the suitable techniques on the problems
CO 5: Understand verification and validation of simulation models.
CO 6: Interpret the models and apply the results to resolve critical issues in a real world environment.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√	√	√				
CO2			√	√			
CO3			√	√	√		
CO4			√	√			
CO5		√	√	√			
CO6				√			√

Module 1: (12 hours)

System models and role of simulation. Entities, Attributes, States and Activities.

Types of systems - Deterministic, Stochastic, Continuous and Discrete systems. Steps in simulation studies.

Statistical tools and techniques- generation of pseudorandom numbers, random variate generation for uniform, Poisson and normal distributions, sampling and estimation, maximum likelihood estimation, confidence intervals and hypothesis testing, stochastic processes and Markov models.

Module 2: (12 hours)

Discrete event simulation languages. Simulation of inventory and queuing systems - single and multiserver queues, network of queues. Modelling and performance evaluation of computers and computer communication networks. Workload characterization.

Module 3: (12 hours)

Continuous system simulation languages, growth and decay models, system dynamics diagrams. Biological and Sociological system simulation. Verification and validation of simulation models - input/output validation, sensitivity analysis, performance measures and their estimation. Case studies.

References:

1. Fishwick P.: Simulation Model Design and Execution, Prentice Hall, 1995.
2. Law A., Kelton D.: Simulation Modelling and Analysis, McGraw-Hill, 1991.
3. Ross, S.: Simulation, Academic Press, 2002.

J. GIS and Remote Sensing

Course Objectives:

- To know the principles, applications, trends, and pertinent issues of geographical information systems and sciences related to GIS.

- To get overview of information retrieval of earth surface features using multi-resolution, multi-scale and multi-temporal imagery.
- To perform image enhancement on remotely sensed imagery and extract information.
- To build critical thinking skills to solve a real-world problem with appropriate remote sensing data.

Course Outcomes:

CO 1: Understand concepts of retrieval of information from remote data.
CO 2: Analyze the energy interactions in the atmosphere and earth surface features.
CO 3: Interpret the images for preparation of thematic maps.
CO 4: Make use of GIS in various engineering applications.
CO 5: Solve spatial problems by analyzing spatial and attribute data.
CO 6: Create GIS and cartographic outputs for presentation.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√		√				
CO2		√		√		√	
CO3		√		√			
CO4	√		√		√	√	
CO5		√		√			
CO6			√	√	√		

Module 1 (12 Hrs)
Fundamentals of remote sensing; Principles of electromagnetic radiation and EM spectrum. Sensors and platforms; remote sensing satellites, multispectral, hyperspectral and thermal sensors; RS data acquisition systems.

Module 2 (10 Hrs)
Image processing; Image enhancement and visualization; Image interpretation and classification. Microwave thermal remote sensing; Radar & laser altimetry.

Module 3 (12 Hrs)
Applications of Remote Sensing; Integration of remote sensing and GIS. Basic concepts of GIS; cartographic principles, map projections and coordinate systems. Geographic information and spatial data types; Hardware and software; Steps of spatial data handling; Database management systems; Spatial referencing.

Module 3 (12 Hrs)
Data quality, measures of location errors on maps. Spatial data input, data preparation; Point data transformation. Analytical GIS capabilities, retrieval and classification, overlay functions. Neighbourhood operations, network analysis, error propagation; Data visualization.

Suggested Books:

1. Burrough PA. Principles of Geographic Information System for Land Resources Assessment. Oxford Univ. Press.
2. Curran PJ. Principles of Remote Sensing. Longman.
3. Jensen JR. Introductory Digital Image Processing. Prentice Hall.

4. Lillesand TM & Kiefer RW. Remote Sensing and Image

K. Internet of Things

Course Objectives:

- Understand the concepts of IoT.
- Study IoT architecture
- Understand the technologies used to build IoT applications.

Course Outcomes:

CO 1: Understand the constraints and opportunities of wireless and mobile networks for Internet of Things.
CO 2: Analyze the IoT enabling technologies.
CO 3: Identify the use of IoT from a global context.
CO 4: Analyze, design and develop parts of an Internet of Things solution and map it toward selected IoT application.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	√			√		√	
CO2	√	√	√	√			
CO3	√	√		√	√	√	√
CO4		√	√	√	√	√	√

Module 1: (12 hours)
Introduction to IoT, Sensing, Actuation, Basics of Networking.
Basics of Networking, Communication Protocols.

Module 2: (12 hours)
Sensor Networks: Machine-to-Machine Communications
Interoperability in IoT, Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino.
Introduction to Python programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi.

Module 3: (12 hours)
Fog Computing, Smart Cities and Smart Homes.
Connected Vehicles, Smart Grid, Industrial IoT.
Case Study: Agriculture, Healthcare, Activity Monitoring.

REFERENCES

1. Peter Waher, "Learning Internet of Things", PACKT publishing, BIRMINGHAM – MUMBAI
2. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", Springer
3. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of

M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications
4. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on Approach)", 1st Edition, VPT, 2014.

L. Real Time System

Course Objectives:

- To understand the applications of Real Time systems.
- To know the concepts of programming in real time environment along with its tools.
- To know the concepts of real time databases.
- Understand the concepts of real time communications and fault tolerance.

Course Outcomes:

CO 1: Demonstrate concepts of Real-Time systems and scheduling.
CO 2: Analyse the programming languages and tools of a real time operational system.
CO 3: Demonstrate the different concepts related to databases in a real time operating system environment.
CO 4: Evaluate the reliability and fault tolerance of a real time model.
CO 5: Understand the concepts of real time communications and fault tolerance.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√	√	√			
CO2			√	√	√		
CO3	√		√			√	
CO4		√	√	√			
CO5		√	√				

Module-1 [8L]
Introduction:
Introduction - Issues in Real Time Computing, Structure of a Real Time System. Task Classes, Performance Measures for Real Time Systems, Estimating Program Run times. Task Assignment and Scheduling - Classical Uniprocessor scheduling algorithms, UniProcessor scheduling of IRIS Tasks, Task Assignment, Mode Changes, and Fault Tolerant Scheduling.

Module-2 [7L]
Programming languages and tools:
Programming Language and Tools – Desired Language characteristics, Data Typing, Control structures, Facilitating Hierarchical Decomposition, Packages, Run-time (Exception) Error handling, Overloading and Generics, Multitasking, Low Level programming, Task scheduling, Timing Specifications, Programming Environments, Run-time Support.

Module-3 [7L]
Real time databases:
Real time Databases - Basic Definition, Real time Vs General Purpose Databases, Main Memory

Databases, Transaction priorities, Transaction Aborts, Concurrency Control Issues, Disk Scheduling Algorithms, Two-phase Approach to improve Predictability, Maintaining Serialization Consistency, Databases for Hard Real Time systems.

Module-4 [7L]
Communication:
Real-Time Communication - Communications Media, Network Topologies Protocols, Fault Tolerant Routing. Fault Tolerance Techniques - Fault Types, Fault Detection. Fault Error containment Redundancy, Data Diversity, Reversal Checks, Integrated Failure handling.

Module-5 [7L]
Evaluation techniques:
Reliability Evaluation Techniques - Obtaining Parameter Values, Reliability Models for Hardware Redundancy, Software Error models. Clock Synchronization - Clock, A Nonfault-Tolerant Synchronization Algorithm, Impact of Faults, Fault Tolerant Synchronization in Hardware, Fault Tolerant Synchronization in Software

Suggested Readings:

1. C.M. Krishna, Kang G. Shin, "Real-Time Systems", McGraw-Hill International Editions, 1997.
2. Stuart Bennett, "Real Time Computer Control-An Introduction",Second edition Perntice Hall PTR, 1994.
3. Peter D. Lawrence, "Real time Micro Computer System Design – An Introduction", McGraw Hill, 1988.
4. S.T. Allworth and R.N. Zobel, "Introduction to real time software design", Macmillan, II Edition, 1987.
5. R.J.A Buhur, D.L. Bailey, "An Introduction to Real-Time Systems", Prentice-Hall International, 1999.
6. Philip.A.Laplante "Real Time System Design and Analysis" PHI , III Edition, April 2004.

M. Image Processing

Course Objectives:

- To know about acquisition, representation and analysis of digital images.
- To learn about image enhancement in spatial domain.
- To comprehend the frequency domain and various image transformation in it.
- To learn the analytical tools and methods applied to image information for various aspects of practical applications.
- To learn image compression and segmentation techniques.

Course Outcomes:

CO 1: Knowledge of basic operators and preprocessing techniques in monochrome and gray images.
CO 2: Skills in image enhancement like linear and non-linear spatial and frequency domain filters using python libraries.
CO 3: Familiarity with concepts and operators of morphological techniques.
CO 4: Knowledge of elemental segmentation algorithms.
CO 5: Understanding about concepts and techniques for image processing projects using different

methods.

CO – PO Mapping:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1		√	√				
CO2	√	√	√	√	√		
CO3	√	√	√	√	√		
CO4	√	√	√	√	√		
CO5	√	√	√	√	√	√	√

Course Content:

Module 1 [8 Hrs]
Introduction: Image Formation, Overview of Image Processing System, Image Digitization (Sampling and Quantization), Digital Image, Fundamentals of Color image, Color Models (RGB, YCbCr, HIS), Image File Format.

Module 2 [6 Hrs]
Image Enhancement in Spatial Domain: linear and non-linear operators, Histogram Equalization, Spatial Domain Smoothing and Sharpening Filters, Correlation and Convolution.

Module 3 [6 Hrs]
Frequency Domain Image Enhancement: Fourier Transform (1-D and 2-D), Frequency Domain image, Image smoothing, Image sharpening, Correlation and Convolution, Discrete Cosine Transform.

Module 4 [6 Hrs]
Morphological Image Processing: Dilation and Erosion, Opening and Closing, Basic morphological algorithms.

Module 5 [10 Hrs]
Image Segmentation, Feature Extraction & Object Recognition: Point Detection, Line Detection, Edge Detection**,** Thresholding, Region segmentation; Chain Codes, Polygonal Approximation, Skeletons, Component Labeling, Texture Analysis, Moments, Gray-level Co-occurrence Matrix, Template Matching, Object recognition methods.

Suggested Books and References:

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”.
2. B. Chanda and D. Dutta Majumder, “Digital Image Processing and Analysis”.
3. S. Sridhar, Digital Image Processing, Oxford University.