PROGRAMME PROJECT REPORT

Bachelor of Computer Application

SCHOOL OF COMPUTER AND INFORMATION SCIENCES
U. P. Rajarshi Tandon Open University
Prayagraj

1. Programme Mission & Objectives

In line with the mission of the University to provide flexible learning opportunities to all, particularly to those who could not join regular colleges or universities owing to social, economic and other constraints, the 3-year Undergraduate Programme in BCA aims at providing holistic and value based knowledge and guidance to promote scientific temper in everyday life. The program offers a platform to the learners to fulfill the eligible criteria in various scientific jobs in government and private sector.

The programme aims at the following objectives:

1. Produce knowledgeable and skilled human resources which are employable in IT and ITES.
2. Impart knowledge required for planning, designing and building complex Application Software Systems as well as provide support to automated systems or application.
3. Produce entrepreneurs who can develop customized solutions for small to large Enterprises.
4. To develop academically competent and professionally motivated personnel, equipped with objective, critical thinking, right moral and ethical values that compassionately foster the scientific temper with a sense of social responsibility.
5. To develop students to become globally competent.
6. To inculcate Entrepreneurial skills among students

2. Relevance of the Programme with Mission and Goals

The 3-year Undergraduate Programme in BCA is designed with the objective of equipping learners to cope with the emerging trends and challenges in the scientific domain. In congruence with goals of the University the Programme also focuses to provide skilled manpower to the society to meet global demands. The Programme is designed with core and elective subjects so that a successful learner can go for higher studies. The Programme also aims at making the learners fit for taking up various jobs.

3. Nature of Prospective Target Group of Learners

The Program is targeted to all individuals looking to earn a graduation degree for employment, further higher education, promotion in career and professional development.

4. Appropriateness of Programme to be conducted in ODL mode to acquire specific skills & competence

Learning outcomes		
Learning Outcomes	**Elements of the descriptor**	**Bachelor of Computer Application**
LO 1	Knowledge and understanding	The graduates should be able to demonstrate the acquisition of comprehensive, factual, theoretical, and specialized knowledge in broad multidisciplinary contexts with depth in the underlying principles and theories relating to one or more fields of learning.
LO 2	Skills required to perform and accomplish tasks	The graduates should be able to demonstrate the acquisition of cognitive and technical skills required for performing and accomplishing complex tasks relating to the chosen fields of learning.
LO 3	Application of knowledge and skills	The graduates should be able to demonstrate the ability to apply the acquired specialized technical or theoretical knowledge, and cognitive and practical skills to gather and analyse quantitative/ qualitative data to assess the appropriateness of different approaches

		to solving problems.
LO 4	Generic learning outcomes	The graduates should be able to demonstrate the ability to communicate in writing and orally the constructs and methodologies adopted for the studies undertaken relating to the chosen fields of learning, make coherent arguments to support the findings/results of the study undertaken to specialist and non-specialist audience.
LO 5	Constitutional, humanistic, ethical and moral values	The graduates should be able to demonstrate the willingness and ability to embrace the constitutional, humanistic, ethical, and moral values, and practice these values in life.
LO 6	Employment ready skills, and entrepreneurship skills and mindset	The graduates should be able to demonstrate the acquisition of • knowledge and essential skills set and competence that are necessary to: take up a professional job and professional practice.

5. Instructional Design

The University follows the credit system in all its programmes. One credit is equal to 30 hours of learner's study time which is equivalent to 15 lectures in conventional system. To earn a bachelor's degree, a learner must earn 120 credits in a minimum of six semesters (three years) with 20 credits per semester. For earning 120 credits, a learner must go through the following Programme Structure:

Programme Structure of BCA

Level	Year	Sem	Core Course 1	Core Course 2	Core Course 3	Core Course 4	Ability Enhancement Compulsory Course (AECC)	Discipline Specific Elective Course (DEC)	Practical Lab/ Project with viva voce	Total credit
5	1	1st	4	4	4	4	2		2	**20**
		2nd	4	4	4	4	2		2	**20**
6	2	3rd	4	4	4	4	2		2	**20**
		4th	4	4	4	4	2		2	**20**
7	3	5th	4	4	4	--	--	4	4	**20**
		6th	4	4	--	--	--	4	8	**20**
			Total credit							**120**

Explanation of terms used for categorization of courses:

A. **Course 1 to 4:** A course, which should compulsorily be studied by a learner as a core requirement is termed as a Core course.

B. **Ability Enhancement Compulsory Courses (AECC):** "AECC" courses are the courses based upon the content that leads to knowledge enhancement.

Semester	Ability Enhancement Courses (AECC)
1	Ability Enhancement Course in English [AECEG] OR Ability Enhancement Course in Hindi [AECHD]
2	Ability Enhancement Course in Human Rights and Duties [AECHRD] OR Ability Enhancement Course in Health & Hygiene [AECHH]
3	Ability Enhancement Course in Environment Awareness [AECEA] OR Ability Enhancement Course in Solid Waste Management [AESWM]
4	Ability Enhancement Course in Nutrition for Community [AECNC] OR Ability Enhancement Course in Disaster Management [AECDM]

C. **Practical Lab:** Lab based on theory courses for implementing the algorithms discussed in theory papers.
D. The learner has to choose any one course from **Discipline Specific Elective Course** in fifth and sixth semester.
E. Project with Viva Voce.

5.1 Course curriculum: The detail of syllabus is given in Appendix-I

5.2 Language of Instruction: English. However, learner can write assignment and give Term End Examination (TEE) either in Hindi or English.

5.3 Duration of the Programme

Minimum duration in years: 03 Maximum duration in years: 06

5.4 Faculty & Support Staff

Professor (1), Assistant Professor (3) and support staff (3)

6. Procedure for admissions, curriculum transaction and evaluation

i. Admission Procedure

(a) The detailed information regarding admission will be given on the UPRTOU website and on the admission portal. Learners seeking admission shall apply online.

(b) Direct admission to 3-year Bachelor of Computer Application program is offered to the interested candidates.

(c) Eligibility:

10+2

OR

3-years diploma from Board of Technical Education / equivalent

OR

Two year ITI programme of any trade after 10 standard.

ii. **Programme Fee:** Rs. 13000/- year. The fee is deposited through online admission portal only.

iii. Evaluation

The evaluation consists of two components: (1) continuous evaluation through assignments, and (2) term-end examination. Learner must pass both in continuous evaluation as well as in the term-end examination of a course to earn the credits assigned to that course. For each course there shall be one written Terminal Examination. The evaluation of every course shall be in two parts that is 30% internal weightage through assignments and 70% external weightage through terminal exams.

(a) Theory course	**Max. Marks**
Terminal Examination	70
Assignment	30
Total	**100**

(b) Practical course:	**Max. Marks**
Terminal Practical Examination	100

Marks of Terminal Practical Examination shall be awarded as per following scheme:

i.	Write up /theory work	30
ii.	Viva-voce	30
iii.	Execution/Performance/Demonstration	20
iv.	Lab Record	20

The following 10-Point Grading System for evaluating learners' achievement is used for CBCS programmes:

10-Point Grading System in the light of UGC-CBCS Guidelines

Letter Grade	Grade Point	% Range
O (Outstanding)	10	91-100
A+ (Excellent)	9	81-90
A (Very Good)	8	71-80
B+ (Good)	7	61-70
B (Above Average)	6	51-60
C (Average)	5	41-50
P (Pass)	4	36-40
NC (Not Completed)	0	0-35
Ab (Absent)	0	
Q	Qualified	Applicable only for Non-Credit courses
NQ	Not Qualified	

Learner is required to score at least a 'P' grade (36% marks) in both the continuous evaluation (assignments) as well as the term-end examination. In the overall computation also, learner must get at least a 'P' grade in each course to be eligible for the B. Sc. degree.

Computation of CGPA and SGPA

(a) Following formula shall be used for calculation of CGPA and SGPA

For jth semester SGPA (Sj) = Σ (Ci *Gi)/ Σ Ci	where, Ci = number of credits of the ith course in jth semester Gi= grade point scored by the learner in the ith course in jth semester.
CGPA = Σ (Cj *Sj)/ Σ Cj	where, Sj = SGPA of the jth semester Cj = total number of credits in the jth semester

The CGPA and CGPA shall be rounded off up to the two decimal points. (For e.g., if a learner obtained 7.2345, then it will be written as 7.23 or if s(he) obtained 7.23675 then it be will written as 7.24)

CGPA will be converted into percentage according to the following formula:

Equivalent Percentage = CGPA * 9.5

(b) Award of Division

The learner will be awarded division according to the following table:

Division	Classification
1st Division	6.31 or more and less than 10 CGPA
2nd Division	4.73 or more and less than 6.31 CGPA
3rd Division	3.78 r more and less than 4.73 CGPA

7. Requirement of the laboratory support and Library Resources

The practical sessions are held in the science laboratories of the Study Centre. In these labs, the learner will have the facility to use the equipment andconsumables relevant to the syllabus. The SLM, supplementary text audio and video material of the various courses of the program is available through the online study portal of the University. The University also have a subscription of National Digital Library to provide the learners with the ability to enhance access to information and knowledge of various courses of the programme.

8. Cost estimate of the programme and the provisions

3-year Bachelor of Computer Application programme consists of 25 theory courses and 05 laboratory and 01 research project. Each course is of 4 credits which consists of approx. 10 units. The total approximated expenditure on the development of 25 courses is:

S. No.	Item	Cost per Unit (writing & editing)	Total cost (Rs.)
1	Total no. of units in 25 courses = 250	4500	11,25,000
2	BOS Meetings etc.	100000	100000
		Total	12,25,000

9. Quality assurance mechanism and expected programme outcomes

(a) **Quality assurance mechanism:** The program structure is developed under the guidance of the Board of studies comprising external expert members of the concerned subjects followed by the School board. The program structure and syllabus is approved by the Academic Council of the University. The course structure and syllabus is reviewed time to time according to the feedback received from the stakeholders and societal needs.

The Centre for Internal Quality Assurance will monitor, improve and enhance effectiveness of the program through the following:

- ✓ Annual academic audit
- ✓ Feedback analysis for quality improvement
- ✓ Regular faculty development programs
- ✓ Standardization of learning resources
- ✓ Periodic revision of program depending upon the changing trends by communicating to the concerned school

(b) **Expected programme outcomes (POs)**

PO1	An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.
PO 2	An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
PO 3	An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
PO 4	An ability to function effectively on teams to accomplish a common goal
PO 5	An understanding of professional, ethical, legal, security and social issues and responsibilities
PO 6	An ability to analyze the local and global impact of computing on individuals, organizations, and society.
PO 7	Recognition of the need for and an ability to engage in continuing professional development.
PO 8	An ability to use current techniques, skills, and tools necessary for computing practice.

APPENDIX-I

Academic Year: 2025-2026

Year wise Structure & Syllabi of Bachelor of Computer Application

Year	Semester	Course Code	Paper Title	Type of Course	Max. Marks	Credits
First	1	**Core Courses**				
		BCA-1.1	C Programming	Theory	100	4
		BCA-1.2	Data Structures	Theory	100	4
		BCA-1.3	Basic Mathematics	Theory	100	4
		BCA-1.4	Numerical Analysis	Theory	100	4
		BCA-1.5P	Practical Work	Practical	100	2
		Ability Enhancement Compulsory Courses				
		AECEG OR AECHD	Ability Enhancement Course in English [AECEG] OR Ability Enhancement Course in Hindi [AECHD]	Theory	100	4
	2	**Core Courses**				
		BCA-2.1	Design and Analysis of Algorithm	Theory	100	4
		BCA-2.2	Discrete Mathematics	Theory	100	4
		BCA-2.3	C++ and Object Oriented Programming	Theory	100	4
		BCA-2.4	Database Management System	Theory	100	4
		BCA-2.5P	Practical Work	Practical	100	2
		Ability Enhancement Compulsory Courses				
		AECHRD OR AECHH	Ability Enhancement Course in Human Rights and Duties [AECHRD] OR Ability Enhancement Course in Health &Hygiene[AECHH]	Theory	100	4
Second	3	**Core Courses**				
		BCA-3.1	Operating System	Theory	100	4
		BCA-3.2	Software Engineering	Theory	100	4
		BCA-3.3	Computer Network	Theory	100	4
		BCA-3.4	Java Programming	Theory	100	4
		BCA-3.5P	Practical Work	Practical	100	2
		Ability Enhancement Compulsory Courses				
		AECEA OR AECSWM	Ability Enhancement Course in Environment Awareness [AECEA] OR Ability Enhancement Course in Solid Waste Management [AESWM]	Theory	100	4
	4	**Core Courses**				
		BCA-4.1	Windows Programming	Theory	100	4
		BCA-4.2	Computer Organization	Theory	100	4
		BCA-4.3	Introduction to Mobile Architecture	Theory	100	4
		BCA-4.4	Introduction to Cyber Security	Theory	100	4
		BCA-4.5P	Practical Work	Practical	100	2
		Ability Enhancement Compulsory Courses				

		AECNC OR AECDM	Ability Enhancement Course in Nutrition for Community [AECNC] OR Ability Enhancement Course in Disaster Management [AECDM]	Theory	100	4
Third	5	**Core Courses**				
		BCA-5.1	Python Programming	Theory	100	4
		BCA-5.2	Multimedia	Theory	100	4
		BCA-5.3	Soft Computing	Theory	100	4
		BCA-5.4P	Practical Work	Practical	100	4
		Discipline Elective Course (select any one)				
		BCA-EA OR BCA-EB	Web Technology OR Client Server Technology	Theory	100	4
	6	**Core Courses**				
		BCA-6.1	Information and Network Security	Theory	100	4
		BCA-6.2	Computer Graphics	Theory	100	4
		BCA-6.3P	Project with Viva Voce	Project	100	8
		Discipline Elective Course (select any one)				
		BCA-EC OR BCA-ED	Computer Architecture OR Microprocessor and its applications	Theory	100	4
Total Max. Marks/Credit					**3300**	**128**

Programme: **BCA**	Year: **First**	Semester: **I**
Subject: BCA		
Course Code: **BCA-1.1**	Course Title: **C Programming**	
Course Objectives: Briefly glimpse the basics of software engineering practices like modularization, commenting, and naming conventions which help in collaborating and programming in teams. This course emphasizes translating any algorithm into C code and writing efficient and maintainable C code.		
Course Outcomes: **CO1** Develop a sound approach to problem solving using C programming language. **CO2** Implement techniques like recursion and iteration are learnt to solve a problem. **CO3** Demonstrate programming concepts like pointers, structures. **CO4** Reading, understanding and modifying code written by others.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Introduction to algorithms and program design**
Unit 1	**Introduction to Algorithms**: Problem solving techniques, Algorithm
Unit 2	**Pseudo-codes and Flowcharts**: Tools of Algorithm, Pseudo codes, Flowchart
Unit 3	**Program design principles**: Introduction to computer programming, Program design principles, Programming techniques, Program Errors
Block 2	**Introduction to the 'C' programming language**
Unit 4	**Introduction:** History of C Language, Structure of a 'C' program, Creating and Executing a 'C' program,
Unit 5	**Data Types in 'C':** Character Set of 'C' language, Trigraph characters, Tokens, Identifiers, Keywords, Constants, Data types, Variables
Unit 6	**Storage Classes:** Scope and lifetime of variable, Storage classes, Automatic storage class, Register storage class, Static storage class, External storage class
Unit 7	**Input and Output Functions:** Reading a single character, Writing a single character, Formatted Input-Output, Formatted Input, Formatted Output
Block 2	**Operator and Control Structures**
Unit 8	**Operators and Expressions:** Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment and decrement operators, Conditional operators, Bitwise operators, Special operators, Operator Precedence and Associativity, lvalue and rvalue, Type casting: Promotion and Demotion of variable types
Unit 9	**Decision Structures in 'C':** if statement, if else statement, nested if … else statement, switch statement, goto statment
Unit 10	**Loop Structures in 'C':** for statement, while statement, do while statement, break statement, continue statement
Unit 11	**Arrays:** One dimensional array, Two dimensional array, Multidimensional arrays, Strings, String handling functions, Character functions
Block 3	**Advanced Features of C**
Unit 12	**Pointers:** Pointers and Address (&) operator, Pointer declaration and Initialization , Indirection operator, Pointer Arithmetic, Arrays and Pointers, Character strings and Pointers, Array of Pointers, Pointer to Pointer
Unit 13	**Functions:** Functions, user-defined functions, categories of function, returning non-integer values, function arguments, recursion, arrays as function arguments
Unit 14	**Structures, Unions, enum and typedef:** Structure definition, Structures within

	structures, Structures as function arguments, Pointers to structures, Unions, Enumerated data type, Type definition
Unit 15	**File and Memory Management in 'C':** Files, File Pointer Variable, Opening a file, Reading and writing to files, File Status Functions, Random Access to files, Command Line Arguments, Memory management
Unit 16	**Preprocessor Directives and Error reporting:** Macro directives, Conditional directives, Control directives, Error reporting

Suggested Readings:

1. Kanetkar, Yashavant. *Let us C*. BPB publications, 2018.
2. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India

Suggested online courses (MOOCs)

1. Introduction to programming in C By Prof. Satyadev Nandakumar, IIT Kanpur
   https://onlinecourses.nptel.ac.in/noc23_cs02/preview
2. Problem Solving Through Programming In C By Prof. Anupam Basu, IIT Kharagpur
   https://onlinecourses.nptel.ac.in/noc23_cs53/preview

Programme: **BCA**	Year: **First**	Semester: **I**
Subject: BCA		
Course Code: **BCA-1.2**	Course Title: **Data Structures**	
Course Objectives: The objective of the course is to familiarize students with basic data structures and their use in fundamental algorithms.		
Course Outcomes: **CO1:** Understand basic data structures such as arrays, strings, and linked lists. **CO2:** Study linear data structures such as stacks and queues and understand their difference. **CO3:** Describe the hash function and concepts of collision and its resolution methods. **CO4:** Study tree, heap and graphs along with their basic operations. **CO5:** Study different techniques for solving problems like sorting and searching		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**BLOCK - 1**
Unit 1	**Introduction to data structure:** Algorithm, Basic criteria for algorithms, Data type, Data structure, Data representation, linear and nonlinear data structure.
Unit 2	**Basics of algorithm:** Algorithm, Basics of complexity of algorithm
Unit 3	**Array:** Definition, Representation of array, Single and multi-dimensional array, address calculation (one dimensional, two dimensional, multidimensional), sparse matrices
Block 2	**BIOCK – 2**
Unit 4	**Stack:** Definition, Operations on stacks, Array representation and implementation of stack; infix, prefix and postfix representation of expression and evaluation multiple stacks, Application of stacks.
Unit 5	**Recursion:** Recursive definition and processes, some named problems of recursion, principle of recursion: designing recursive algorithm, how recursion works, tail recursion.
Unit 6	**Queue:** Definition, operation on queues, circular queue, dequeue, priority queue, Application of queue.
Block 3	**BLOCK 3**
Unit 7	**Linked List:** Representation and implementation of single linked list, Operations in the singly linked list, stack and queue as a linked list, circularly linked list, doubly linked list, circularly doubly linked list, Application of linked list: polynomial representation and addition, garbage collection
Unit 8	**Tree:** Basic terminology, binary tree, binary tree representation, complete binary tree, extended binary tree, array and linked list representations, traversing binary tree, threaded binary tree, binary search tree, Operations on BST, AVL tree, Operations on AVL tree, B-tree Insertion and deletion in B tree.
Unit 9	**Graph:** Basic terminology Graph representation Depth first search, breadth first search, topological sort, connected components, spanning tree, minimum cost spanning tree, Kruskal's and prim's algorithm, Shortest path algorithms: Bellman Ford Algorithm, Dijkstra's algorithm, Floyd-Warshall algorithm.
Block 4	**BLOCK- 4**
Unit 10	**Searching and sorting:** Sequential search, binary search, comparison and analysis, Selection sort, Bubble sort, Insertion sort, Heap sort, Quick Sort, Merge sort, Shell sort, radix sort.
Unit 11	**Hashing:** Hash table, hash function, collision resolution strategies, hash table implementation.
Unit 12	**File Structure:** Terminology, File organization, Sequential files, Direct File organization, Indexed Sequential file organization.

Suggested Readings:

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
2. R.L. Kruse: Data Structures & Program Design in C, PHI.

Suggested online courses (MOOCs)

1. Programming and Data Structure, IIT Kharagpur by Dr. P.P.Chakraborty
   https://nptel.ac.in/courses/106105085
2. NOC:Programming and Data structures (PDS), IIT Madras by Dr. N S. Narayanaswamy
   https://nptel.ac.in/courses/106106130
3. NOC:Programming, Data Structures and Algorithms, IIT Madras by Prof. Hema A Murthy, Dr. N S. Narayanaswamy, Prof. Shankar Balachandran
   https://nptel.ac.in/courses/106106127
4. Data Structures And Algorithms, IIT Delhi by Prof. Naveen Garg
   https://nptel.ac.in/courses/106102064

This course can be opted as an elective by the students of following subjects: N.A.

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **First**	Semester: **I**
Subject: BCA		
Course Code: **BCA-1.3**	Course Title: **Basic Mathematics**	
Course Objectives: The course offers an introduction to basic mathematics which is essentials for a computer science student to go ahead and study any other topics in the subject. The emphasis will be on problem solving as well as proofs. This course focuses on Set Theory, Functions, Limits and Continuity, Quadratic Equation and Calculus.		
Course Outcomes: **CO1** Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions. **CO2** Find limits of functions. **CO3** Analyze and apply the notions of continuity and differentiability to algebraic and transcendental functions.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Unit 1	**Sets &. Relations :** Sets and elements, Equal sets, Universal set & Empty set, Subsets, Venn diagrams, Basic operations on sets, Union & Intersection, Complements, Difference, Symmetric Difference, Fundamental Products, Algebra of sets and Duality, Finite Sets, Counting Principle, Classes of sets, Power sets, Partitions, Mathematical Induction, Cartesian Products of Sets, Relations, Pictorial representations of Relations, Composition of relations, Types of relations, Equivalence Relations, Partial ordering relations.
Unit 2	**Functions, Limits and Continuity** : Functions, Kinds of Functions , Concept of real function, Domain and Range (simple cases), Composition Function, One-to-one, onto, into, invertible functions, Mathematical Functions , Exponential and Logarithmic Functions, Graph of functions (plotting of linear function, absolute value function, parabolic functions, Sin(x), Cos(x), tan(x), reciprocal function, ex, log x, Signum function), Polar coordinates and graph, Limit of variable, Limit of function, Evaluation of limits of various types of functions, Continuity & Discontinuity at a point, Continuity over an interval. Trigonometrical Functions: Definitions, proofs for any angle 0, signs of ratios, ratios of some standard angles.
Unit 3	**Quadratic Equation:** Solution of Quadratic Equations, Nature of Roots. Co-ordinates and Loci: Cartesian co-ordinate system, Introduction to Polar co-ordinates, distance between two points, section formulae, Area of triangle, Locus and its Equation. Straight Line: Equation of straight line parallel to an Axis, slope form, intercept form, through two point condition of concurrency of three lines. Matrices and Determinants : Definition and Types of Matrices, Addition , Subtraction and Multiplication of a Matrices, Scalar Multiplication, Transpose of Matrix, Determinants, Determinants of square matrix of order I, 2 and 3, Area of a triangle, Solution of system of linear equations by Cramer's Rule, Minors and Cofactors, Adjoint of a Matrix, Inverse of a Matrix(up to order 3).
Unit 4	**Differential Calculus:** Derivative of a Function, Various Formulae-Product and Quotient Rule of Differentiation, Differentiation of Function of Function(chain rule), Trigonometrical functions, Inverse Trigonometrical functions, Exponential function, Logarithmic function, Implicit functions, Logarithmic Differentiation, Differentiation of function with regard to another function, Higher Derivatives, Successive Differentiation, Liebnitz Theorem, Expansion of functions(up to 3 or 4 terms only) using Maclaurin's and Taylor's 1 theorem, Maxima and Minima (simple cases), Curve tracing (simple cases), Introduction to partial differentiation.
Unit 5	**integral Calculus :** Anti-Derivatives, Constant of integration, Indefinite integral, Elementary Integration, formulae, Methods of Integration, Integration by Substitution,

	Integration by parts, integration through partial fractions and rationalization, Concept of Definite integral, properties of definite integral, Integration using Gamma function. Area of Bounded Region, Circle, Parabola, Ellipse in standard form between two coordinates and x- axis.
Suggested online courses (MOOCs) 1. Basic Calculus 1 and 2 By Prof. Parasar Mohanty, IIT Kanpur https://onlinecourses.nptel.ac.in/noc21_ma20/preview	
This course can be opted as an elective by the students of following subjects: N.A.	
Suggested equivalent online courses (MOOCs) for credit transfer: N.A	

Programme: **BCA**	Year: **First**	Semester: **I**
Subject: BCA		
Course Code: **BCA-1.4**	Course Title: **Numerical Analysis**	
Course Objectives: The course provides students with an understanding to develop numerical methods for various mathematical problems and calculate the error involved in the numerical solution when compared to their exact solution. These techniques are useful to students while solving various interdisciplinary science problems. It contains solution of system of linear equations, roots of non-linear equations, interpolation, numerical differentiation and integration.		
Course Outcomes: **CO1** Understand basic methods and principles of scientific computing. **CO2** Solve basic and frequently occurring mathematical problems using computers and numerical software.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	**Solutions of Non-Linear Equations in one Variable**	
Unit 1	Review of Calculus, Round off Error, Truncation Error, Some properties of equations, Iteration Methods for finding the roots (zero's) of an equation. Convergence Criterion, Initial Approximation to a Root, Bisection Method,	
Unit 2	Fixed Point Iteration Method, Chord Methods for Finding Roots- Regula Falsi Method, Newton Raphson Method. Order of convergence.	
Block 2	**Solution of System of Linear Equations**	
Unit 3	Direct Methods- Preliminaries, Method of solution using inverse of matrix. Cramer's rule. Gauss Elimination Method, Gauss- Jordon Reduction Method, LU decomposition method. Crout's method.	
Unit 4	Iterative Method- General Iteration Method, Jacobi's Iteration Method, Gauss- Seidal Iteration Method, Relaxation method.	
Block 3	**Interpolation**	
Unit 5	Definition, Finite Differences: Forward differences, Backward differences, Central differences, Other differences operator, Relation between operators. Interpolation at Equally interval; Newton Gregory formula for forward differences and backward difference.	
Unit 6	Interpolation at Unequally interval Lagrange's interpolation formula. Divided differences, Properties of divided differences, Newton's Divided difference interpolation formula.	
Block 4	**Numerical Differentiation, Integration and Solutions of Differentiation Equations**	

Unit 7	Numerical Differentiation, Numerical Integration; Trapezoidal Rule. Simpson's One Third Rule, Simpson's Three Eight's Rule. Weddle's Rule.
Unit 8	Numerical Solution of Ordinary Differential Equations-(first order, second order and simultaneous) by Picard's Iteration Method, Euler's Method, Runge- Kutta Methods- 4^{th} Order.

Suggested Readings:

1. K. E. Atkinson, An Introduction to Numerical Analysis, 2nd Edition, John Wiley, 2008.
2. Numerical Analysis, R. L. Burden and J. D. Faires, 7th ed., Thomson Learning, 2001.

Suggested online courses (MOOCs)

1. NOC:Numerical methods, IIT Roorkee by Prof. Ameeya Kumar Nayak, Prof. Sanjeev Kumar https://nptel.ac.in/courses/111107105
2. NOC:Numerical Methods for Engineers, IIT Madras by Dr. Niket S.Kaisare https://nptel.ac.in/courses/127106019
3. Numerical Analysis By Prof. S. Baskar, IIT Bombay https://onlinecourses.nptel.ac.in/noc23_ma44/preview

This course can be opted as an elective by the students of following subjects: **B.Sc.(Computer Science), M.Sc. (Statistics) and M.Sc. (Mathematics)**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **First**	Semester: **I**
Subject: Computer Science		
Course Code: **BCA-1.5P**	Course Title: **Practical Work**	

Course Objectives:

- Develop logical and algorithmic thinking to solve computational problems.
- Provide hands-on experience in writing, debugging, and executing C programs.
- Introduce and implement fundamental data structures for efficient data organization.
- Apply numerical methods to solve mathematical and engineering problems computationally.
- Promote good programming practices, teamwork, and problem-solving skills in real-world applications

Course Outcomes:

CO1 Design algorithms and develop structured solutions using C programming.
CO2 Implement programs using arrays, structures, and file handling with proper error checking.
CO3 Apply and analyze basic data structures such as stacks, queues, linked lists, trees, and graphs.
CO4 Use numerical methods to solve equations, perform interpolation, differentiation, integration, and solve ODEs.
CO5 Demonstrate effective debugging, documentation, and teamwork skills in software development tasks.

Credits: **02**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in C Programming:

1. Design an **algorithm**, write the **pseudo-code**, and draw a **flowchart** to check whether a given number is **prime or not**. Identify possible **program errors** during implementation.
2. Write, compile, and execute a C program to display personal details (name, roll number, CGPA) using **different data types, constants, and variables**.
3. Write a program to demonstrate the behavior of **automatic, static, register, and external** storage classes and explain their **scope and lifetime**.
4. Develop a program that takes user input for **marks in three subjects**, calculates the **total and average**, and displays results in a **formatted table** using `printf()` and `scanf()`
5. Write a program that reads three numbers and determines the **largest number** using **if–else** statements and **relational/logical operators**. Modify it to use the **conditional operator**.
6. Create a program to input `n` integers into an array, then use **loops** to find the **maximum, minimum, and average**, and display the array in **reverse order**.
7. Write a program that uses a **structure** to store student details (roll number, name, marks), saves data for five students into a **file**, and then reads and displays the contents with **error handling**.

List of Practical in Data Structures:

1. Design and implement algorithms to find the **maximum and minimum element in an array**, and analyze their **time and space complexity**.
2. Implement operations on single and multi-dimensional arrays, and develop a program to store and add sparse matrices efficiently.
3. Write a program to implement a **stack using arrays**, perform **push and pop** operations, convert an

infix expression to postfix, and **evaluate** it.

4. Implement **recursive algorithms** (factorial and Fibonacci) and a **queue using arrays**. Extend the program to demonstrate **circular and priority queues**.
5. Develop programs to perform basic operations on **singly, doubly, and circular linked lists**, and implement **polynomial addition** using linked lists.
6. Create and traverse a **binary tree** using inorder, preorder, and postorder traversals, and perform **insertion, deletion, and searching** in a **binary search tree**.
7. Implement **graph representation** (adjacency matrix and list) and perform **BFS** and **DFS** traversals. Also implement and compare **two sorting algorithms** (e.g., Quick Sort and Merge Sort).

List of Practical in Data Structures:

1. Write a program to find a real root of $f(x) = x^3 - 4x - 9 = 0$ using the Bisection Method with a tolerance of 10^{-4}. Display the approximate root and number of iterations.
2. Implement Fixed Point Iteration, Newton-Raphson, and Regula Falsi methods for $f(x) = e^x - 3x = 0$. Compare their convergence rates and the number of iterations required for tolerance 10^{-5}.
3. Solve the following system using Gauss Elimination, Gauss-Jordan, and LU Decomposition (Crout's Method): $2x + y - z = 8$; $-3x - y + 2z = -11$; $-2x + y + 2z = -3$. Display the final values of x, y, z.
4. Use Jacobi, Gauss-Seidel, and Relaxation Method (ω=1.1) to solve equations $4x - y + z = 7$, $-2x + 6y + z = 9$ and $x + y + 5z = -6$. Compare the number of iterations for convergence (error < 10^{-4}).
5. Given x: 0,1,2,3,4 and f(x): 1,3,7,13,21, find f(2.5) using Newton-Gregory Forward Difference Formula and Lagrange's Interpolation Formula.

6. For f(x) = sin(x) on [0, π/2] with step h = π/10:

(a) Compute f'(x) at x = π/4 using forward and central difference.

(b) Evaluate ∫₀^{π/2} sin(x) dx using Trapezoidal, Simpson's 1/3, and Simpson's 3/8 rules.

7. Solve dy/dx = x + y, y(0) = 1, for x = 0 to 1 with h = 0.1 using:

(a) Euler's Method (b) Improved Euler (Heun's) Method (C) Runge-Kutta 4th Order Method.

Suggested Readings:

1. Uttar Pradesh Rajarshi Tandon Open University, BCA-1.1 *C programming: Self-learning material.* School of Sciences, UPRTOU, 2020.
2. Uttar Pradesh Rajarshi Tandon Open University, *BCA-1.2 Data structure: Self-learning material.* School of Computer and Information Sciences, UPRTOU, 2020.
3. Uttar Pradesh Rajarshi Tandon Open University, *BCA-1.4 Numerical Analysis: Self-learning material.* School of Sciences, UPRTOU, 2021.
4. Virtual Lab on Data Structure: https://ds1-iiith.vlabs.ac.in/

Programme: **BCA**	Year: **First**	Semester: **II**
Subject: Computer Science		
Course Code: **BCA-2.1**	Course Title: **Design And Analysis Of Algorith**	

Course Objectives: This course provide the common paradigms to design efficient algorithms for real world problem solving. It gives an understanding of how to analyze the asymptotic performance of algorithm; write rigorous correctness proofs for algorithms; important algorithmic design paradigms and methods of analysis; efficient algorithms in common engineering design situations.

Course Outcomes:

CO1 Understand that various problem solving methods exist such as; iterative technique, divide and conquer, dynamic programming, greedy algorithms.

CO2 Analyze the strengths and weaknesses of an algorithm theoretically as well as practically.

CO3 Identify and apply an appropriate technique to design an efficient algorithm for simple problems.

CO4 Demonstrate correctness and efficiency of the algorithm.

CO5 Apply various searching and sorting algorithms.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Introduction and Design Strategies-I**
Unit 1	**Introduction:** Algorithm, Psuedo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Growth of functions: Asymptotic Notation, Recurrences: substitution method, master method.
Unit 2	**Divide and Conquer:** General method, applications-Binary search, Finding the maximum and minimum, Quick sort, Heapsort, Strassen's Matrix Multiplication.
Unit 3	**Sorting in Linear Time:** Lower bounds for sorting, Counting sort, Radix sort, Bucket sort, Medians and Order Statistics, Minimum and maximum.
Block 2	**Algorithm Design Strategies-II**
Unit 4	**Greedy method:** General method, applications- Knapsack problem, Job sequencing with deadlines, optimal two way merge patterns, Huffman codes, Minimum cost spanning trees: Prims and Kruskal's algorithm, Single source shortest paths: The Bellman-Ford algorithm, Dijkstra's algorithm.
Unit 5	**Dynamic Programming:** General method, applications, capital budgeting problem, Multistage graphs, Matrix chain multiplication, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem.
Block 3	**Algorithm design strategies & Completeness**
Unit 6	**Graph Algorithms:** Introduction, representation of graphs, Breadth first search, depth first search, topological sort, strongly connected component, flow networks, ford-fulkerson method.
Unit 7	**Backtracking:** General method, applications, 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.
Unit 8	**Branch-And-Bound:** The method, travelling salesperson problem, 15 puzzle problem.
Unit 9	**NP-Hard and NP-Complete problems:** Basic concepts, non deterministic algorithms, NP - Hard and NP Complete classes, satisfiability problem, reducibility.

Suggested Readings:

1. Cormen, Leiserson, Rivest,and Stein, "Introduction to Algorithms", MIT Press ,Third Edition, 2009.
2. Dasgupta, Papadimitrou and Vazirani, "Algorithms", McGraw-Hill Education, 2006. Horowitz, Sahni, and Rajasekaran, "Computer Algorithms" Silicon Press, 2007

Suggested online courses (MOOCs)

1. NOC:Design and Analysis of Algorithms, Chennai Mathematical Institute By Prof. Madhavan

Mukund https://nptel.ac.in/courses/106106131 2. NOC:Introduction to algorithms and analysis, IIT Kharagpur by Prof. Sourav Mukhopadhyay https://nptel.ac.in/courses/106105164 3. Design and Analysis of Algorithms, IIT Bombay By Prof. Abhiram Ranade https://archive.nptel.ac.in/courses/106/101/106101060/#
This course can be opted as an elective by the students of following subjects: **MCA**
Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **First**	Semester: **II**
Subject: **BCA**		
Course Code: **BCA-2.2**	Course Title: **Discrete Mathematics**	
Course Objectives: This course provides students understand discrete objects such as proofs, sets, graphs, colorings, algebraic structures and algorithms that arise naturally and frequently in many areas of mathematics and computer science. It develops a sound understanding of these discrete objects to solve problems arising in computer science.		
Course Outcomes: **CO1** Apply mathematical logic to solve problems. **CO2** Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, and functions. **CO2** Understand and apply counting techniques to the representation and characterization of relational concepts. **CO2** Impart foundations of probabilistic theory which is mostly used in varied applications in engineering and science.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Language of Mathematics and its application**
Unit 1	**Mathematical Logic:** statements, operations, truth values, tautology and quantifiers.
Unit 2	**Arguments:** Rule of Detachment, Validity of a compound statement by using Truth Table, Validity using Simplification Methods, Validity using Rules of Inference, Invalidity of an Argument, Indirect Method of proof and Proof by Counter-Example.
Unit 3	**Boolean Algebra:** Boolean Algbera, Principle of Duality, Isomorphic Boolean Algebras, Boolean Algebra as Lattices, Boolean Functions, Disjunctive Normal Form**,** Conjunctive Normal Form**,** Minimization of Boolean Functions (Karnaugh Map)
Unit 4	**Switching circuits and logical Circuits:** Switching Circuits, Simplification of circuit, Non-Series Parallel Circuits, Relay Circuits, Logic Circuits
Block 2	**Set theory and its application**
Unit 5	**Set theory:** sets, Subsets, Operations on Sets, Complementation, Intersection and Union, Laws Relating Operations, Distributive Laws and De Morgan's Laws.
Unit 6	**Relation:** Relation, binary relations in a Set, Domain and Range of a Relation, Total number of Distinct Relations, Relations as Sets of Ordered Pairs, Types of Relations, Composition of Relations, Equivalence relation in a set, Partition of a Set, Equivalence Class and Quotient set of a set.
Unit 7	**Partitions and Distributions:** Equivalence Relations, Equivalence Classes, Properties of Equivalence Classes, Quotient set and Partition.
Unit 8	**Function:** Functions, Direct and Inverse image, Inverse Functions, Operations on Functions, Composite of functions, Types of Functions and Connection between Equivalence relation and mapping.
Block 3	**Counting Process**
Unit 9	**Mathematical Induction:** Principle of Mathematical Induction, Second Principle of Induction and Well ordering property.
Unit 10	**Combinatorics:** Basic counting principles**,** Principle of Disjunctive counting, Principle of Sequential counting and Ordered and Unordered Partitions.
Unit 11	**Permutation**

Unit 12	**Combination**
Block 4	**Block – 04: Probability theory and application**
Unit 13	**Binomial theorem:** Binomial theorem, General term in a binomial expansion,Middle term in a binomial expansion and Binomial expansion for rational exponents.
Unit 14	**Probability:** Definition of Probability, Addition law for counting and Product law for counting.
Unit 15	**General Counting methods:** General Counting method is the extension part of counting process. It discusses Sum and Product Rules and the Pigeonhole Principle**.**
Unit 16	**The Inclusion- Exclusion Principle:** inclusion-exclusion principle, Alternative form of the inclusion-exclusion principle and Onto Functions.

Suggested Readings:

1. C.L.Liu and D.P.Mohapatra, " Elements of Discrete Mathematics: A Computer Oriented Approach", Mcgraw Hill, Third Edition, 2012.
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications" Mcgraw Hill, Seventh Edition, 2012 (Indian Adaptation by Kamala Krithivasan, Iit Madras).

Suggested online courses (MOOCs)

1. NOC:Discrete Mathematics, IIT Ropar, Prof. Prabuchandran K.J, Prof. Sudarshan Iyengar; https://nptel.ac.in/courses/106106183
2. NOC:Discrete Mathematics, IIT Guwahati, Prof. Benny George K, Prof. Sajith Gopalan https://nptel.ac.in/courses/106103205

This course can be opted as an elective by the students of following subjects: **B.Sc. in Computer Science, B.Sc. in Physics, B.Sc. in Statistics.**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **BCA**	Year: **First**	Semester: **II**
Subject: BCA		
Course Code: **BCA-2.3**	Course Title: **C++ and Object-oriented programming**	
Course Objectives: This course aims to offer a practical mastery of object-oriented concepts such as classes, objects, data abstraction, methods, method overloading, inheritance and polymorphism.		
Course Outcomes: **CO1** Develops a sound approach to problem solving using a middle level programming language. **CO2** Apply techniques like recursion and iteration are learnt to solve a problem. **CO3** Build programming concepts like pointers, structures.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**BLOCK - 1**
Unit 1	**Principles of object-oriented programming:** Object oriented programming paradigm, Comparison with procedural programming, Basic concepts of object-oriented programming, benefits of O0P, object-oriented Languages, advantage of C++.
Unit 2	**Object Orient Programming System:** Class, inheritance, abstraction, encapsulation and information hiding, polymorphism, overloading.
Unit 3	**Advanced concept**: Dynamism (Dynamic typing, dynamic binding, late binding, dynamic loading). Structuring programs, reusability, organizing object-oriented project,
Block 2	**BIOCK - 2**
Unit 4	**Overview of C++:** Tokens, keywords, identifiers and constants basic data types, user-defined and derived Data types, type compatibility, reference, variables type Casting, operator precedence, control structures, structure, function.
Unit 5	**Classes and objects:** Class specification, class objects, accessing class members, scope resolution operator, data hiding, empty classes, Pointers within a class, passing objects as arguments, returning objects from functions, friend Functions and friend classes, constant parameters and member functions, structures and Classes, static members.
Unit 6	**Object initialization and cleanup**: Constructors destructor, constructor overloading, order of construction and destruction, Constructors with default arguments, nameless objects, dynamic initialization through Constructors, constructors with dynamic operations, constant objects and constructor, static Data members with constructors and destructors, nested classes.
Block 3	**BLOCK - 3**
Unit 7	**Operator overloading and type conversion:** Defining operator overloading, overloading unary operators, overloading binary operators, overloading binary operators using friends, manipulation of strings using Operators, rules for overloading operators. type conversions.
Unit 8	**Inheritance: extending classes:** Deriving derived classes, single multilevel, multiple, hierarchical, hybrid inheritance, Constructors & destructors in derived classes, constructors invocation and data members Initialization, virtual base classes, abstract classes, delegation.
Block 4	**BLOCK- 4**
Unit 9	**Pointers, virtual functions and polymorphism:** Pointers to objects, this pointer. pointers to derived classes, virtual functions, Implementation of run-time polymorphism, pure virtual functions.
Unit 10	**Working with files:** Classes for file stream operations. opening and closing a file, file pointers and their Manipulations, sequential input and output operations, error handling during file Operations, command line arguments.

Unit 11	**Object Oriented Modeling:** Need of object-oriented Modeling, Simulation of real-life problems using OOP concept: Example, Representation of problem using object and class diagrams at design level.

Suggested Readings:

1. The C++ Programming Language by Bjarne Stroustrup, 2013.
2. Programming: Principles and Practice Using C++ by Bjarne Stroustrup, 2014
3. Oriented Object-Oriented Programming with C++ by Balaguruswamy, TMH

Suggested online courses (MOOCs)

1. NOC:An Introduction to Programming Through C++, IIT Bombay by Prof. Abhiram G Ranade https://nptel.ac.in/courses/106101208
2. Programming in Modern C++, IIT Kharagpur By Prof. Partha Pratim Das https://onlinecourses.nptel.ac.in/noc23_cs50/preview

This course can be opted as an elective by the students of following subjects: N.A.

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **First**	Semester: **II**
Subject: BCA		
Course Code: **BCA-2.4**	Course Title: **Data Base Management System**	

Course Objectives: Today databases form the backbone of all major applications – internet, banking, product & sales etc. Relational Database Management Systems (DBMS) have long formed the basis for many leading databases such as Oracle, Microsoft SQL Server and MySQL. This course aim to provide a common set of models and design paradigms which includes:

- Data models, conceptualize and depict a database system using ER diagram.
- Internal storage structures in a physical DB design.
- Database normalization technique that organizes the data within a database in the most efficient manner possible.
- Fundamental concepts of transaction processing techniques**.**

Course Outcomes:

CO1 Students can explain the role of a database management system, basic database concepts, including the structure and operation of the relational data model.

CO2 Apply logical database design principles, including E-R/EE-R diagrams, conversion of ER diagrams to relations.

CO3 Describe the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.

CO4 Construct simple and moderately advanced database queries using Structured Query Language (SQL).

CO5 Understand and apply Database Normalization to remove the duplicate data and database anomalies from the relational table

CO6 Understand the concept of a database transaction including concurrency control, backup and recovery.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Basic concepts of DBMS**
Unit 1	**Introduction: Database** Management System, Examples, Characteristics of the Database Approach, Advantage of using a Database Approach. Database System concepts and Architecture, Data Models, Schemes and Instances, DBMS Architecture and Data independence, Database Languages, Procedural and Non-procedural languages and Interfaces. Database System Environment, Classification of Database Management Systems.
Unit 2	**ER Model:** Database Modeling using the ER Model., Using High-Level conceptual Data Models for Database design, an example Database Application, Entity types, Entity Sets, Attributes and keys, Relationships, Relationship types, roles and Structural Constraints., Week Entity types, Refining the ER Design for the Company Database, ER Diagrams, naming conventions and design Issues, Conversion of ER Diagram to tables.
Unit 3	**Relational Data Model:** Basic Relational data model Concepts, Relational Databases and Relational Database Schemas, Relational Model Constraints, update Operations and Dealing with Constraint Violations
Block 2	**Query Language and Database Design Concepts**
Unit 4	**Relational Algebra:** Relational Model Concepts, Relational concepts and Relational Database Schemes, Update Operation and Dealing with Constraints Violations, Relational Database Design, Using ER-to-Relational Mapping.
Unit 5	**Structured Query language:** Data definition, Constraints and Schema changes in SQL 2, Basic Quires in SQL, More Complex SQL Quires, Insert, Delete and Update Statements in SQL, views (Virtual Tables) in SQL, Specifying general constraints as

	Assertion features of SQL. Integrity constraints, Triggers, Functional dependencies.
Unit 6	**Functional Dependency Theory: Functional** Dependencies and Normalization for Relational Database, Informal Design Guidelines for Schemes, Functional Dependencies.
Unit 7	**Normalization:** Normal Forms based on Primary keys, General Definitions of Second and Third Normal forms, Boyce Codd Normal form, Relational Database Design Algorithms and Further Dependencies, Algorithms for Relational Database Schema Design, Multi-valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Inclusion Dependencies, Other Dependencies and Normal Forms.
Block 3	**Transaction Management & Emerging Databases**
Unit 8	**Transaction Processing Concepts**: Introduction to Transaction Processing, Transaction and System Concept, Desirable properties of Transactions, Scheduling and Recoverability, Serializability of Scheduling, Transaction Support in SQL, Concurrency control techniques, Concurrency techniques for concurrency control, concurrency control based on timestamp based protocol, validation based protocol, deadlock handling, Database Recovery Techniques based on Immediate Update, Failure classification, Shadow Paging, Log based recovery, failure with loss of Nonvolatile Storage.
Unit 9	Emerging Trends in DBMS: Emerging Trends in DBMS: Introduction to object-oriented Database Management System, Introduction to client/Server Database, Introduction to Distributed Database, Introduction to Knowledge Databases.

Suggested Readings:

1. R Elmasri, S Navathe, Fundamentals of Database Systems, 6th edition, Addison-Wesley, 2010.
2. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd Ed., McGraw-Hill, 2002.
3. A Silberschatz, H Korth and S Sudarshan, Database System Concepts, 6th Ed., McGraw-Hill, 2010.

Suggested online courses (MOOCs)

1. NOC: Data Base Management System, IIT Kharagpur by Prof. Partha Pratim Das Prof. Samiran Chattopadhyay Prof. Kausik Datta
   https://nptel.ac.in/courses/106105175
2. NOC:Introduction to Database Systems, IIT Madras by Prof. P.Sreenivasa Kumar
   https://nptel.ac.in/courses/106106220
3. NOC:Fundamentals of Database Systems (Course sponsored by Aricent), IIT Kanpur By Dr. Arnab Bhattacharya
   https://nptel.ac.in/courses/106104135

This course can be opted as an elective by the students of following subjects: **B.Sc. in Computer Science, MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **First**	Semester: **II**
Subject: BCA		
Course Code: **BCA-2.5P**	Course Title: **Practical Work**	

Course Objectives:

- To understand the fundamental concepts of database systems, object-oriented programming, numerical methods, and algorithm design.
- To develop skills in designing databases using ER modeling, SQL implementation, relational algebra, and normalization techniques.
- To enhance programming proficiency in C++ through OOP concepts including classes, inheritance, polymorphism, operator overloading, and file handling.
- To strengthen algorithmic problem-solving skills by implementing divide-and-conquer, greedy, dynamic programming, backtracking, and graph algorithms.
  - To apply theoretical knowledge to practical scenarios involving transactions, concurrency, ACID properties, and performance analysis of algorithms.

Course Outcomes:

CO1 Design and implement relational databases, including ER diagrams, relational tables, SQL queries, views, triggers, and normalization to ensure data integrity and reduce redundancy.

CO2 Develop C++ programs demonstrating OOP principles such as encapsulation, inheritance, polymorphism, operator overloading, constructors/destructors, dynamic initialization, and file handling.

CO3 Analyze and implement classical and advanced algorithms using divide-and-conquer, linear-time sorting, greedy, dynamic programming, backtracking, branch-and-bound, and graph algorithms.

CO4 Apply transaction management concepts in databases, ensuring ACID properties, handling deadlocks, and understanding recovery and serializability mechanisms.

CO5 Demonstrate practical problem-solving skills by modeling real-world problems using OOP, algorithmic strategies, and numerical methods, and evaluate the efficiency and correctness of solutions.

Credits: **02**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

1. Consider a **University Database** with entities such as Student, Faculty, Course, and Department. Identify entities, attributes (primary and foreign keys), and relationships and draw an **ER Diagram** representing the database. Convert your ER Diagram into **relational tables** with defined keys.
2. Create the tables derived from earlier using SQL `CREATE TABLE` statements. Define primary keys, foreign keys, and integrity constraints. Insert sample data into the tables and demonstrate update, delete, and insert operations, and handle constraint violations.
3. Write SQL queries to:
   - List all students enrolled in a particular course.
   - Find faculty teaching more than 3 courses.
   - Display departments having more than 10 students.
   - Retrieve students who do not belong to any department.

Create a **view** for students' grades in all courses and define a **trigger** to automatically update a student's status when their GPA falls below a threshold.

4. Consider the tables created earlier, express the following queries using **relational algebra**:
   - Find all students enrolled in "Database Systems" taught by "Prof. X".
   - List departments and the total number of students in each department.
   - Retrieve faculty teaching courses with no students.

Perform join, projection, selection, union, difference operations.

5. For the `Course_Registration(Student_ID, Student_Name, Course_ID, Course_Name, Faculty_Name, Department)` table, identify the functional dependencies, then normalize it through 1NF, 2NF, 3NF, and BCNF. Explain how normalization enhances data integrity and minimizes redundancy.
6. Simulate a bank database with an `Account(Account_ID, Name, Balance)` table and write SQL transactions to transfer money between accounts, ensuring ACID properties with `COMMIT` and `ROLLBACK`. Demonstrate a deadlock scenario, explain its resolution, and discuss how serializability and recovery mechanisms apply to these transactions.

List of practical in numerical methods and OOP with C++:

1. Write a C++ program demonstrating classes and objects, showcasing data encapsulation with private and public members, and compare its results with a procedural version of the program.
2. Implement a Bank Account class with member functions for deposit, withdrawal, and display, utilize friend functions and pass objects as arguments, and demonstrate returning objects from functions.
3. Implement a Student class with constructors and destructors, demonstrating constructor overloading, dynamic initialization, static data members, and nameless objects, while tracking the order of construction and destruction through output statements.
4. Create a Complex number class with overloaded +, −, *, and == operators, and demonstrate unary operator overloading along with type conversion between the class and primitive types.
5. Implement a base class `Employee` with derived classes `Manager` and `Engineer`, demonstrating single, multilevel, and multiple inheritance, constructors and destructors in derived classes, virtual base classes, and how data members are initialized through constructors.
1. Create a base class Shape with virtual and pure virtual functions for area() and perimeter(), derive classes Circle, Rectangle, and Triangle, and use base-class pointers to invoke virtual functions, demonstrating runtime polymorphism.
2. Implement a Student Record Management System using files, performing operations such as open,

read, write, sequential I/O, and error handling, and represent the system with class and object diagrams at the design level.

List of practical in Design and Analysis of Algorithm:

3. Write pseudo-code for linear and binary search, analyze their time and space complexities, demonstrate growth of functions with examples of $O(n)$, $O(n^2)$, and $O(\log n)$, and solve a recurrence relation using the substitution method and the master theorem.
4. Implement in C/C++ binary search, finding maximum and minimum in an array, quick sort, and Strassen's matrix multiplication using divide and conquer, and compare their performance with simple iterative approaches.
5. Implement in C/C++ Counting Sort, Radix Sort, and Bucket Sort, find the minimum, maximum, and median of a list of numbers, and analyze and compare the running times of all algorithms.
6. Implement in C/C++ greedy algorithms for the 0/1 Knapsack problem (or fractional if allowed), job sequencing with deadlines, Huffman coding for data compression, and Prim's or Kruskal's algorithm for minimum spanning trees, and compare the results with optimal solutions where feasible.
7. Implement in C/C++ dynamic programming algorithms for the 0/1 Knapsack problem, Matrix Chain Multiplication, and the all-pairs shortest path problem (Floyd-Warshall), demonstrating intermediate tables or memoization and reconstruction of the optimal solutions.
8. Implement graph traversal algorithms—BFS, DFS, and topological sort—and solve backtracking problems including the 8-queen problem, sum of subsets, and Hamiltonian cycle detection, representing graphs using both adjacency lists and adjacency matrices.

Programme: **BCA**	Year: **Second**	Semester: **III**
Subject: BCA		
Course Code: **BCA-3.1**	Course Title: **Operating System**	

Course Objectives: The course will introduce Operating Systems (OS), their design and implementation. We will discuss the goals of an OS and some successful and not-so-successful OS designs. We will also discuss the following OS services in detail: thread scheduling, security, process management, memory management, virtual memory, and disk scheduling.

Course Outcomes:

CO1 Analyze & classify different types of operating system

CO2 Understand the working of Operating system

CO3 Interpret concepts of thread scheduling, process management, memory management, virtual memory, and disk scheduling.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**An Overview and Process Management**
Unit 1	**Introduction:** Basic definitions, Batch processing, Multi-programming. Time sharing, multiprocessing; Structure and Functions of Operating System
Unit 2	**Process and thread:** Process, Process states, State Transitions, Process Control Block, Context Switching, concept of thread, comparison between process and thread, Thread model, thread usage, implementing thread in kernel and user space.
Unit 3	**Process Scheduling:** Scheduler, Scheduling criteria, Preemptive and non-preemptive scheduling, Process Scheduling, Process scheduling algorithms.
Unit 4	**Concurrent Process:** Process Interaction, Shared Data and Critical Section, Mutual Exclusion, Synchronization, Classical Problems of Synchronization, Semaphores, Monitors.
Block 2	**Memory Management and Unix Case Study**
Unit 5	**UNIT 5: Deadlock:** Concept of deadlock, necessary condition for deadlock, resource allocation graph, deadlock prevention, deadlock avoidance, Banker's algorithm, Deadlock detection, deadlock recovery.
Unit 6	**UNIT 6: Memory management:** Address Binding, Dynamic Loading and Linking Concepts, Logical and Physical Addresses Contiguous and non-contiguous memory allocation, Paging, Segmentation, Virtual Memory, Demand Paging, Page fault,Page replacement algorithms, thrashing.
Unit 7	**UNIT 7: Secondary memory management:** Free Space management, Disk Structure, Disk Scheduling, Formatting, Swap space Management.
Unit 8	**UNIT 8: Case Study of UNIX**

Suggested Readings:

1. Silberschatz, Galvin, Gagne, Operating System Concepts, 8th Edition, Wiley,2008
2. Andrew S. Tanenbaum, Albert S. Woodhull, Operating Systems, Design and Implementation, 3rd Edition, Prentice Hall,2006.
3. William Stallings, Operating Systems: Internals and Design Principles, 6th Edition, Prentice Hall, 2013.
4. Charles Patrick Crowley, Operating Systems-A Design-oriented Approach. 1996

Suggested online courses (MOOCs)

1. NOC:Operating System Fundamentals, IIT Kharagpur by Prof. Santanu Chattopadhyay https://nptel.ac.in/courses/106105214
2. NOC:Introduction to Operating Systems, IIT Madras by Prof. Chester Rebeiro https://nptel.ac.in/courses/106106144
3. Operating Systems, IIT Delhi by Prof. Sorav Bansal https://nptel.ac.in/courses/106102132

Programme: **BCA**	Year: **Second**	Semester: **III**
Subject: BCA		
Course Code: **BCA-3.2**	Course Title: **Software Engineering**	
Course Objectives: Provide the current software engineering techniques and examine the software life-cycle, including software specification, design implementation, testing and maintenance. It presents software engineering methodologies for the development of Quality, cost-effective, schedule meeting software.		
Course Outcomes: **CO1** Describe software engineering layered technology and process framework. **CO2** Introduces theories, models, and techniques that provide a basis for the software development life cycle. **CO3** Introduces software testing approaches including verification and validation, static analysis, reviews, inspections, and audits. **CO4** Understanding of the role of project management including planning, scheduling, risk management, etc. **CO5** Work as an individual and/or in team to develop and deliver quality software.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Unit 1	**Software Engineering Fundamentals:** Definition of Software, Software characteristics, Software Applications. Software Process: Software Process Models - Waterfall model, prototyping model, spiral model, incremental model, concurrent development model. Project management Concepts: The Management Spectrum - The People, The Product, The Process, The Project.
Unit 2	**Software Process and Project Metrics :** Measures , Metrics and Indicators , Software measurement Size -Oriented Metrics , Function - Oriented Metrics , Extended Function point metrics Software Project Planning : Project Planning Objectives , Software Project Estimation , Decomposition Techniques - Problem Based Estimation Process Based Estimation ,Empirical Estimation Models- The COCOMO Model Risk Analysis and Management: Software risks, Risk identification, Risk Projection, Risk Refinement, Risk Mitigation , Monitoring and Management.
Unit 3	**Software Quality Assurance:** Basic concepts- Quality, Quality Control, Quality Assurance, Cost of Quality, Software Quality Assurance (SQA), Formal Technical Review Software Configuration Management: Baselines, Software Configuration Items, The SCM Process, Version Control, Change Control, Configuration Audit, Status Reporting. Analysis Concepts and Principles: Requirements Elicitation for Software, Analysis Principles. The Information Domain, Modeling, Partitioning, Essential and Implementation Views, Specification: Specification Principles, Representation, The Software Requirement Specification (SRS)
Unit 4	**Design Concepts and Principles:** Design Principles, Design Concepts — Abstraction, Refinement, Modularity, Software Architecture, Control Hierarchy, Structural Partitioning, Data Structure. Software Procedure, Structure, Information Hiding, Effective Modular Design- Cohesion, Coupling Software Testing: Testing Objectives & principles, Unit Testing, Integration Testing (Top-Down Integration, Bottom. Up Integration, Regression Testing, Smoke Testing), Validation Testing (Alpha and Beta Testing), System Testing (Recovery Testing, Security Testing, Stress Testing, Performance Testing).
Unit 5	**Reengineering:** Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering CASE Tools: What is CASE, Building Blocks of CASE, A Taxonomy of CASE Tools, Integrated CASE Environments, The integration Architecture, The CASE Repository.

Suggested Readings:

1. Mall, Rajib. Fundamentals of software engineering. PHI Learning Pvt. Ltd., 2018.
2. R.S. Pressman, Software Engineering – A Practitioner's Approach, 6th Edition, TMH, 2013.
3. Ian Sommerville, Software Engineering, 8th Edition, Addison Wesley, 2009.
4. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing, 2010.

Suggested online courses (MOOCs)

1. NOC:Software Engineering, IIT Kharagpur by Prof. Rajib Mall https://nptel.ac.in/courses/106105182
2. Software Engineering, IIT Bombay by Prof. Rushikesh K Joshi, Prof. Umesh Bellur, Prof. N.L. Sarda https://nptel.ac.in/courses/106101061

This course can be opted as an elective by the students of following subjects: N.A.

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **Second**	Semester: **III**
Subject: BCA		
Course Code: **BCA-3.3**	Course Title: **Computer Networks**	
Course Objectives: This course offers students an understanding of how machines are connected in a network and how data communication takes place between machines at various locations. It provides basic concepts of data communication, layered model, protocols and interworking between computer networks and switching components in telecommunication systems.		
Course Outcomes: **CO1** Understand basics of computer networks and various network topologies. **CO2** Explain basics of OSI Reference Model and TCP/IP Model. **CO3** Understand various protocol of data link layer for flow and error control such as Stop and wait protocols, One bit sliding window protocol, Using Go-Back N. **CO4** Describe different types of network devices Hub, Bridges, Switch, Gateways, and Routers along with their working. **CO5** Aware of different types of IP addresses classes and the need of subnetting. **CO6** Realize how packet is being transferred from source to destination PC. **CO7** Examine different types of routing protocols, flow control, error control and congestion control algorithms in network and transport layer.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	Computer Network Basics and Services
Unit 1	Introduction to Computer Network: Computer networks, Network Hardware—Local Area networks, Metropolitan Area networks,,Wide Area networks, Wireless networks, Internetworks, Network Software: Protocol Hierarchies,
Unit 2	OSI and TCP/IP Model: Design and Issue for layers, Interfaces and services, Connection oriented and Connection less Services. OSI reference model, and its Evolution, TCP/IP model.
Unit 3	The Physical Layer: Physical Layer, Transmission media, twisted pair, Base band and Broadband coaxial cable, Fiber optics, unguided media.
Unit 4	ISDN and Switching Techniques: MODEM, ISDN services, Switching Message, Packet Circuit switching TDM, and FDM, ATM, X.25.
Block 2	Link Layer Issues and Access Protocols
Unit 5	Unit 5: Data Link Layer: Data Link Layer, Error detection and Correction, Protocols: Simplex Stop and wait protocols, One bit sliding window protocol, Using Go-Back N. Flow control, Sliding Window Protocol, Channel Allocation Problem,
Unit 6	Unit 6: Multiple Access Protocol: ALOHA, CSMA protocol, Collision Free protocol, Polling, FDM, TDM,
Unit 7	Unit 7: The Medium Access Sub Layer: Framing, Static and Dynamic Channel Allocation in LANs and MANs, IEEE Standard 802.3, and Ethernet IEEE standard 802.4 and token Ring, IEEE Standard 802.5, Token Bus
Unit 8	Network devices: Hub, Bridges, Switch, Gateways, Routers.
Block 3	IP Addressing and Routing Issues
Unit 9	IP Protocol and Addressing: Network layer design issue, IP Protocol, IP Addresses, subnets,
Unit 10	Connection Management: Internetworking, connection less and connection oriented services, tunneling, Fragmentation, Firewall, Internet Controls Protocols.
Unit 11	Routing in Network Layer: Routing Algorithm, shortest path routing, Flooding, Flow-based routing, Broadcast routing, Congestion Control Algorithm, Congestion control and

	prevention policies;
Block 4	Transport, Session, Presentation and Application Layer
Unit 12	Transport layer: Transport layer connection management, flow control, error control, congestion control, Establishing and releasing a connection, TCP service Model, TCP protocol
Unit 13	Session and Presentation Layer: Introduction to cryptography and data compression
Unit 14	The Application Layer: Network Security, Domain Name System, Email: Architecture and Services, Message formats, Message transfer.

Suggested Readings:

1. HBehrouz A. Forouzan, Data Communications and Networking, McGraw Hill , 2006
2. A.S. Tanenbaum, Computer Networks, PHI , 2002

Suggested online courses (MOOCs)

1. Data Communication, IIT Kharagpur by Prof. Ajit Pal
   https://nptel.ac.in/courses/106105082
2. NOC:Computer Networks and Internet Protocol, IIT Kharagpur by Prof. Soumya Kanti Ghosh, Prof. Sandip Chakraborty
   https://nptel.ac.in/courses/106105183
3. NOC:Advanced Computer Networks, IIT Indore, IIT Gandhi nagar by Prof. Neminath Hubballi, Prof. Sameer Kulkarni
   https://nptel.ac.in/courses/106106243

This course can be opted as an elective by the students of following subjects: **B.Sc. in computer science, MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **Second**	Semester: **III**
Subject: BCA		
Course Code: **BCA-3.4**	Course Title: **Java Programming**	
Course Objectives: This course aims to cover the essential topics of Java programming so that students can improve their skills to cope with the current demand of IT industries and solve many problems in their field of study.		
Course Outcomes: **CO1** Use the characteristics of an object-oriented programming language JAVA in a program. **CO2** Apply JAVA features to program design and implementation. **CO3** Design and implementation programs of Java Script, Applets, Event Handling, AWT Programming, and Interface. **CO4** Implementation of Packages, Swing, and Servlet. **CO5** Design and implementation programs of JSP.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	**Object Oriented Methodology and Java**	
Unit 1	**Object Oriented Programming:** Paradigms of Programming languages, Evolution of Object-Oriented Methodology, Basic Concepts of OO Approach, Comparison of object oriented and procedure - oriented Approaches, Benefits of OOPS, Applications of OOPS. Classes and objects, Abstraction and Encapsulation, Inheritance, Method overriding and Polymorphism.	
Unit 2	**Java Language Basics**: Introduction to Java, Primitive Data Type and Variables, Java Operators.	
Unit 3	**Expressions Statements and Arrays:** Expressions, Statements, Control Statements, Selection Statements, Iterative Statements, Jump statements, Arrays.	
Block 2	**Object oriented concepts and Exceptions Handling**	
Unit 4	**Class and objects:** Class Fundamentals, Introducing Methods, this Keyword, Using objects as Parameters, Method overloading, Garbage collection, the ffinalize () Method.	
Unit 5	**Inheritance and Polymorphism:** Inheritance Basics, Access, Multilevel, inheritance, Method overriding Abstract classes, Polymorphism, Final Keyword.	
Unit 6	**Packages and interfaces:** Package, Accessibility of Packages, using Package members, Interfaces, Implementing interfaces, interface and Abstract classes, Extends and Implements together.	
Unit 7	**Exceptions Handling:** Exception, Handling of Exception, Types of Exceptions, Throwing, Exceptions, writing Exception subclasses.	
Block 3	**Multithreading, I/O, and Strings Handling**	
Unit 8	**Multithreaded Programming:** Multithreading, The Main thread, JAVA Thread Model, Thread Priorities, Synchronization in JAVA, Inter thread Communication.	
Unit 9	**I/O In Java:** I/O Basics, Streams and stream, Classes, the predefined streams, Reading from and writing to console, reading and writing files, the transient and volatile Modifiers, using instance of Native Methods.	
Unit 10	**Strings and Characters:** Fundamental of Characters and Strings, the String class, String operations, Data Conversion using value of () Methods, Strings Buffer and Methods.	
Unit 11	**Exploring Java I/O:** Java I/O classes and interfaces, Stream classes, Text streams, Stream Tokenizer, Serialization, Buffered stream, print stream, Random Access file.	
Block 4	**Graphics and user interfaces**	
Unit 12	**Applets:** The applet class, Applet architecture, An applet Skeleton: Initialization and Termination, Handling events, HTML Applet TAG.	
Unit 13	**Graphics and user interfaces:** Graphics contests and Graphics objects, user interface components, Building user interface with AWT, Swing - Based GUI, Layouts and layouts	

	and layout Manager, Container.
Unit 14	**Networking Features:** Socket overview, reserved parts and proxy servers, Internet Addressing: Domain Naming Services (DNS), Java and The Net: URL, TCP/IP Sockets, Datagrams.
Suggested Readings: 1. Java: The Complete Reference Hebert Schildt, Mc Graw Hill 2. Object-Oriented Programming with C++ and Java Debasis Samanta, Prentice Hall India. **Suggested online courses (MOOCs)** 1. NOC:Programming in Java, IIT Kharagpur by Prof. Debasis Samanta: https://nptel.ac.in/courses/106105191	
This course can be opted as an elective by the students of following subjects: **B.Sc. in computer science**	
Suggested equivalent online courses (MOOCs) for credit transfer: N.A	

Programme: **BCA**	Year: **Second**	Semester: **III**
Subject: Computer Science		
Course Code: BCA-3.5P	Course Title: **Practical Work**	

Course Objectives:

- To develop a strong understanding of **object-oriented programming principles** using Java, including abstraction, encapsulation, inheritance, and polymorphism.
- To enable students to design and implement **modular, reusable, and efficient Java programs** using packages, interfaces, exceptions, threads, and file handling.
- To strengthen students 'ability to **develop GUI-based and interactive applications** using AWT, Swing, and Applets for real-world problem-solving.
- To impart practical skills in **operating system fundamentals**, including process management, scheduling, synchronization, memory management, deadlock handling, and file systems through C/C++ simulations.

Course Outcomes:

CO1 Apply **object-oriented programming concepts** to design, develop, and test Java programs effectively.

CO2 Implement decision-making, looping, exception handling, and multithreading mechanisms in Java to create robust and efficient applications.

CO3 Demonstrate the use of file handling, serialization, and string manipulation for data persistence and text processing in Java.

CO4 Develop and deploy graphical and event-driven Java applications using AWT, Swing, and Applets.

CO5 Simulate and analyze key **operating system functions** such as process scheduling, deadlock avoidance, memory allocation, and disk scheduling using C/C++ to understand system-level behavior and efficiency

Credits: **02**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in Java Programming Lab:

1. Write Java programs demonstrating classes, objects, abstraction, encapsulation, inheritance, and polymorphism, and implement method overloading and overriding with suitable examples.
2. Develop programs using primitive data types, operators, and expressions, implement decision-making with if-else and switch, use loops (for, while, do-while) for iteration, and work with 1D and 2D arrays for computations such as matrix addition or searching.
3. Create user-defined packages and interfaces and demonstrate their use across multiple classes, implement abstract classes and interfaces together in a small application (e.g., Shape or Vehicle system), and write Java programs to handle checked and unchecked exceptions using try-catch-finally, throw, and throws.
4. Create multiple threads using both the Thread class and Runnable interface, implement thread synchronization to prevent race conditions, and demonstrate inter-thread communication using `wait()`, `notify()`, and `notifyAll()`.
5. Develop a Java program that demonstrates file handling by reading from and writing to files using `FileInputStream`, `FileOutputStream`, `BufferedReader`, and `PrintWriter`; implement object serialization and deserialization to store and retrieve object data; and perform various string operations such as concatenation, comparison, modification, and reversal using the `String`, `StringBuffer`, and `StringBuilder` classes.
6. Design a Java application that demonstrates graphical user interface development by creating a simple AWT or Swing-based program such as a student registration form or calculator, and develop

an interactive applet incorporating event handling mechanisms, displaying it through an HTML `<applet>` tag.

List of Practical in Operating System Lab:

1. Write a C/C++ program to create multiple processes using `fork()` or equivalent, displaying process IDs, parent-child relationships, and process states; implement threads using pthreads (C/C++), and compare execution and resource sharing between processes and threads.
2. Simulate in C/C++ preemptive scheduling (Round Robin) and non-preemptive scheduling (FCFS, SJF) for a set of processes with given burst and arrival times, compute waiting time, turnaround time, and average times, and compare the efficiency of the algorithms on the same processes.
3. Implement in C/C++ mutual exclusion using semaphores or mutex, solve the Producer-Consumer and Reader-Writer problems with semaphores or monitors, and demonstrate critical section protection and race condition avoidance.
4. Simulate in C/C++ a system with multiple processes and resources, and implement the Banker's algorithm for deadlock avoidance.
5. Simulate in C/C++ contiguous and non-contiguous memory allocation, simulate page replacement algorithms (FIFO, LRU, Optimal), and demonstrate page faults and thrashing using a sample process sequence.
1. Simulate in C/C++ disk scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN), implement free space management using a linked list or bitmap, and demonstrate basic UNIX commands for file handling, process monitoring, and memory inspection.

Suggested Readings:

1. https://www.cdlsiet.ac.in/wp-content/uploads/2022/03/DBMS-LAB-MANUAL.pdf
2. https://mrcet.com/pdf/Lab%20Manuals/CSE%20II-II%20SEM.pdf

Programme: **BCA**	Year: **Second**	Semester: **IV**
Subject: BCA		
Course Code: **BCA-4.1**	Course Title: **Windows Programming**	
Course Objectives: This course intended to provide Windows programming and its associated concepts like traditional programming, programming resources, and visual C++ programming. It discusses about the Visual Basic programming concepts including the controls available in control/tool box and other custom controls. It imparts understanding of the document view architecture concepts using MFC, Single Document Interface (SDI), Multiple Document Interface (MDI), Database Management System (DBMS), Network programming concepts, ActiveX Controls, COM, DCOM, and COM+.		
Course Outcomes: **CO1** Understand basics of visual basic and its various components. **CO2** Apply event driven model and object oriented methodology. **CO3** Use basic programming skills using GUI interfaces to develop various applications.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Introduction to Windows Programming**
Unit 1	Windows Programming, Traditional Programming Paradigms, Event Driven Programming, Handles and Data Types, Windows Messages, Device Contexts, Document Interfaces, Dynamic Linking Libraries, Software Development Kit Tools, Context Help
Unit 2	Programming Resources: Accelerators, Bitmaps, Dialog Boxes, Icons, Menus, String Tables, Toolbars.
Unit 3	Visual C++ Programming: Basic Concepts of VC++, Object Oriented Programming, Objects and Classes, VC++ Components, Resources, Event Handling, Menus, Dialog Boxes, MFC File Handling, MFC and VC++
Block 2	**Visual Basic Programming**
Unit 4	Visul Basic Programming: Introduction to Visual Basic, Important Windows, Variables, Data Types, Decision Making, Operators, Loops, Procedures in Visual Basic, Visual Basic Code Module
Unit 5	Working with Controls: What is Control, What is Custom Control, Control Properties, The Intrinsic Controls, RichTextBox Controls, Working with Menu Items, Adding and Removing Control, Naming a Control
Unit 6	Dialog Boxes and Internet: Introduction to Dialog Boxes, Modal Dialog Box, Modeless Dialog Box, Modal Vs Modeless Dialog Box, Common Dialog Box, Visual Basic and Internet
Block 3	**Working with Graphics**
Unit 7	Document View Architecture: Microsoft Foundation Class, View Document Architecture Using MFC, Serialization, Separating documents from view, Visual C++ Resources, Application Wizard, Accelerators, Menus, Toolbars
Unit 8	Graphics and Multimedia: Working with Graphics, Consoles, Multitasking Process and Threads, Drawing Graphics in Windows, Clipboards, Printing Graphics and Text, Creating Animations with Picture Clip control
Block 4	**Block 4 Interfacing and Database Application**
Unit 9	Interfacing Other Applications: Single Document Interface (SDI), Multiple Document Interface (MDI), Difference between SDI & MDI, Explorer Style-Interface, Splitter Windows, Exception Handling, Debugging, Object Linking and Embedding (OLE)
Unit 10	Database Application: History of DBMS, Introduction to DBMS, DBMS Architecture, Components of DBMS, Need of DBMS, Advantages of DBMS, Disadvantages of DBMS, Database Administrator (DBA), Open Database Connectivity (ODBC), Database Access, Structured Query Language (SQL), Database Access with Data Control, Recordset, Applications of DBMS

Unit 11	Network Programming: Introduction to Winsock, Windows Socket in General, Creating Sockets, Miscellaneous API, Winsock Catalog, Windows Objects, Access Control Story, Security Descriptors
Unit 12	Advance Topics and Case Study: ActiveX Control, Component Object Model (COM), COM+, Distributed Component Object Model (DCOM), Application using Visual Basic, Example - Customer Database Input Screen

Suggested Readings:

1. Starting Out with Visual Basic, 7th Edition. Gaddis. Addison-Wesley. ISBN: 978-0134400150.
2. Programming Windows with MFC, Second Edition, by Jeff Prosise
3. Wiley India VB.Net Step By Step, Michael Halvorson, PHI.

This course can be opted as an elective by the students of following subjects: **B.Sc.(Computer Science), M.Sc. (Statistics) and M.Sc. (Mathematics)**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **Second**	Semester: **IV**
Subject: BCA		
Course Code: **BCA-4.2**	Course Title: **Computer Organization**	

Course Objectives: The course aims to provide an understanding of the basic structure of a digital computer and to study the operations of internal components.

Course Outcomes:

CO1 Assess basics components of computer hardware.

CO2 Understand how Boolean algebra is related to designing computer logic, through simple combinational and sequential logic circuits.

CO3 Realize a simple computer with hardware design including data format, instruction format, instruction set, addressing modes, bus structure, input/output, memory, Arithmetic/Logic unit, control unit, and data, instruction and address flow.

CO4 Design combinational and sequential logic circuits, flip-flops, counters, shift registers, adders, substractor, multiplexer, demultiplexer, Arithmetic/Logic unit.

CO5 Develop concept of memory unit and input/output architecture.

CO6 Build basics of Instruction Set Architecture (ISA).

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Introduction to Digital Electronics**
Unit 1	**Introduction to number system:** binary, octal, hexadecimal, Inter-conversion to different number system.
Unit 2	**Boolean algebra and Logic Gates:** De Morgan's theorem, Boolean Identity. OR, AND NOT NAND, NOR and Ex OR gates and their Truth Tables, Positive and Negative logic.
Unit 3	**Reduction Techniques:** Standard representation of Boolean expressions, SOP and POS forms, Combinational and sequential circuits, Minterm and Maxterm expressions, Map reduction techniques, K- tap. Code Conversions: Binary to Gray, BCD to decimal etc.
Unit 4	**Binary Arithmetic:** Half and Full Adder, Substractor, Multiplexer, Demultiplxer, Decoder, Encoders, Comparators.
Unit 5	**Sequential Circuit:** Flip Flops: S/R, J/K, D and T Latches, Digital Counters, Registers.
Block 2	**Basic building blocks**
Unit 6	**Building blocks:** I/O, Memory, ALU and its components, Control Unit and its functions
Unit 7	**Instruction** — word, Instruction and Execution cycle, branch, skip, jump and shift instruction, Operation of control. registers; Controlling of arithmetic operation.
Unit 8	**Addressing techniques —** Direct, Indirect, Immediate, Relative, Indexed addressing and paging. Registers —Indexed, General purpose, Special purpose, overflow, carry, shift, scratch, Memory Buffer register; accumulators; stack pointers; floating point; status information and buffer registers.
Block 3	**Memory & I/O**
Unit 9	**Memory**: Main memory, RAM, static and dynamic, ROM, EPROM, EEPROM, EAROM, Cache and Virtual memory.
Unit 10	**I/O System:** Buses, Interfacing buses, Bus formats- address, data and control, Interfacing keyboard, display, auxiliary storage devices and printers.
Unit 11	**Introduction to Microprocessors and microcontrollers**; Introduction to 8085 microprocessor, example of few instructions to understand addressing techniques, differences between microprocessors and microcontrollers. Interlocution to different processor families.

Suggested Readings:

1. William Stallings, "Computer Organization and Architecture", 9th Edition, PHI,2012

2. M. Morris Mano, Michael D. Ciletti, "Digital Design", 4th Edition, Pearson Education, 2011.
3. Hennessy J. and Patterson D., "Computer Architecture: A Quantitative Approach", 5th Edition, Morgan Kaufmann, 2011.

Suggested online courses (MOOCs)

1. Digital Computer Organization, IIT Kharagpur by Prof. P.K. Biswas
   https://nptel.ac.in/courses/117105078
2. NOC:Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof. Kamalika Datta
   https://nptel.ac.in/courses/106105163
3. NOC:Computer Organization and Architecture, IIT Madrasby Prof. V. Kamakoti
   https://nptel.ac.in/courses/106106166
4. Computer Organisation and Architecture, IIT Kanpurby Prof. Bhaskaran Raman
   https://nptel.ac.in/courses/106104073

This course can be opted as an elective by the students of following subjects: **B.Sc. in computer science**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **Second**	Semester: **IV**
Subject: BCA		
Course Code: **BCA-4.3**	Course Title: **Introduction to Mobile Architecture**	

Course Objectives:

- To introduce students to the fundamentals, architecture, and challenges of mobile application development across various platforms.
- To understand the design principles, user interface considerations, and testing practices for creating efficient and user-friendly mobile apps.
- To explore the architecture and features of major mobile operating systems such as Android, iOS, and Windows.
- To examine the hardware components of mobile devices—processors, memory, and sensors—that impact application design and performance.

Course Outcomes:

CO1 Explain the **concepts, evolution, and architecture** of mobile applications and operating systems.
CO2 Design and analyze **mobile app interfaces and functionalities** following industry best practices and security standards.
CO3 Compare the **features and architectures** of Android, iOS, and Windows mobile platforms.
CO4 Evaluate how **mobile hardware components** influence application performance, resource management, and user experience.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT**
Unit 1	**INTRODUCTION TO MOBILE APPLICATIONS:** Introduction, Considerations and Challenges for Mobile App, PC Based Applications, Web Based Applications, Evolution of Mobile Based Apps, Comparison of Mobile App with Web Application, Content and Protocol in Mobility, Trends in Mobility Space, Brief note on Mobile App Platforms
Unit 2	**Components of a Mobile Application:** Introduction, Architecture of a Mobile Application, Components of a Mobile Client Application, Components of Mobile Support Infrastructure, End to End Case Study of Android Mobile Architecture
Unit 3	**Basics of Mobile Application Design:** Introduction, Design Considerations and Best Practices, Checklist for Mobile Apps, User Interface Design for Mobile Apps, Deployment, Power Usage, Synchronization, Patterns and Design Elements, Security Standards and Best Practices, Mobile App Testing
Block 2	**MOBILE OPERATING SYSTEMS**
Unit 4	**Introduction to Mobile Operating Systems:** Introduction, Basic Functions of an Operating System, Mobile Operating Systems, Architecture of Android, Knowing the Operating System of a Mobile Phone, Discontinued Mobile Operating Systems, Existing Mobile Operating Systems, Types of Mobile Operating Systems
Unit 5	**Basics of Android:** Introduction, Interface, Applications, Memory Management, Virtual Reality
Unit 6	**Basics of iOS:** Introduction, Accessibility, Multitasking, Switching Applications, Ending Tasks, Siri, Setting up Siri, Launching Siri, Game Center
Unit 7	**Basics of Windows Mobile:** Introduction, Development, Evolution of Windows Phone, Features of Windows Phone, Virtual Private Networking, Releases, Windows Phone 7, Windows Phone 8, Windows 10 Mobile
Block 3	**MOBILE HARDWARE**

Unit 8	**Mobile Processors:** Introduction, Mobile Processors, Qualcomm Snapdragon, Samsung Exynos, NVIDIA Tegra, More Mobile Processors, ARM Processors, Features of ARM processor, ARM architecture, x86 Processors, Basic Design of x86 Processor, Instruction Execution Cycle, Differences Between x86 and ARM Processors
Unit 9	**Memory:** Introduction, Memory in a Mobile Phone, Volatile Memory, Non-Volatile Memory, Memory Card, ROM, MROM, PROM, EPROM, EEPROM, Flash Memory, NOR Memories, NAND Memories
Unit 10	**Sensors:** Introduction, Gyroscope, Accelerometer, Specification of an Accelerometer, Output of an Accelerometer, Applications of an Accelerometer, Compass, Proximity Sensor
Unit 11	**Input-Output:** Introduction, Display, Camera, Speakers, Active Speakers, Passive Speakers, Mic

Programme: **BCA**	Year: **Second**	Semester: **IV**
Subject: BCA		
Course Code: **BCA-4.4**	Course Title: **Introduction to Cyber Security (OER UoU)**	

Course Objectives:

- To introduce the fundamental concepts of cyber space, cyber crimes, and malware, along with their classifications and impact.
- To familiarize students with cyber security mechanisms such as authentication, encryption, digital signatures, antivirus, firewalls, and forensic investigation.
- To provide awareness of national cyber security initiatives, password management, and secure configuration practices for systems and networks.
- To develop knowledge and safe practices for secure browsing, social media usage, wireless communication, and smartphone protection.

Course Outcomes:

CO1 Explain the **nature of cyber crimes** and identify various forms of malware and their preventive measures.

CO2 Apply **cyber security tools and techniques** such as encryption, authentication, and firewalls to protect systems and data.

CO3 Demonstrate **safe internet and social media practices**, including secure browsing, password protection, and privacy management.

CO4 Evaluate and implement **security measures** for wireless networks and smartphones to ensure data confidentiality and user safety.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	
Unit 1	**INTRODUCTION TO CYBER SPACE:** INTRODUCTION, INTRODUCTION TO CYBER CRIME, MALWARE AND ITS TYPE, KINDS OF CYBER CRIME
Unit 2	**CYBER SECURITY TECHNIQUES:** AUTHENTICATION, ENCRYPTION, DIGITAL SIGNATURES, ANTIVIRUS, FIREWALL, STEGANOGRAPHY
Unit 3	**INVESTIGATING CYBER CRIMES: INTRODUCTION TO CYBER FORENSIC:** COMPUTER FORENSICS, WHY SHOULD WE REPORT CYBER CRIME?
Unit 4	**SOME RECENT CYBER SECURITY ATTACKS:** INTRODUCTION, SOME RECENT CYBER CRIME INCIDENTS
Block 2	
Unit 5	**CYBER SECURITY INITIATIVES IN INDIA:** NTRODUCTION, COUNTER CYBER SECURITY

	INTIATIVES IN INDIA
Unit 6	**GUIDELINES FOR SECURE PASSWORD, TWO STEP VERIFICATION AND USING FREE ANTIVIRUS:** GENERATING SECURE PASSWORD, Guideline for setting secure Password, USING PASSWORD MANAGER, ENABLING TWO-STEP VERIFICATION, SECURING COMPUTER USING FREE ANTIVIRUS
Unit 7	**CONFIGURING FIREWALL IN YOUR COMPUTER:** CONFIGURING FIREWALL ON MAC COMPUTER, Turning on and Configuring the Mac OS X Firewall, WORKING WITH WINDOWS FIREWALL IN WINDOWS7, Firewall in Windows 7, Configuring Windows Firewall, How to Start & Use the Windows Firewall with Advanced Security
Unit 8	**CHOOSING BEST BROWSER TO SUIT YOUR REQUIREMENTS:** FINDING THE BEST BROWSER ACCORDING TO THE USERS REQUIREMENT
Block 3	
Unit 9	**GUIDELINES FOR SAFE INTERNET BROWSING:** SAFE BROWSING, How do I know if a website is secure?, TIPS FOR BUYING ONLINE, CLEARING CACHE FOR BROWSERS
Unit 10	**WIRELESS SECURITY:** WHAT IS WIRELESS LAN?, MAJOR ISSUES WITH WLAN, Secure WLAN, Wi-Fi at home
Unit 11	**EMAIL AND SOCIAL MEDIA SECURITY:** SAFE BROWSING GUIDELINES FOR SOCIAL NETWORKING SITES, General Tips on using Social Networking platforms safely, Posting Personal Details, Friends, Followers and Contacts, Status Updates, Sharing Online Content, Revealing your Location, Sharing Videos and Photos, Instant Chats, Joining and Creating Groups, Events and Communities, EMAIL SECURITY TIPS,
Unit 12	**SMARTPHONE SECURITY:** INTRODUCTION, SMARTPHONE SECURITY GUIDELINES, Purses, Wallets, Smartphones, Platforms, Setup and Installation, Communicating Securely(Through Voice and Messages) with a Smartphone,

Programme: **BCA**	Year: **Second**	Semester: **IV**
Subject: BCA		
Course Code: **BCA-4.5P**	Course Title: **Practical Work**	

Course Objectives:

- To develop proficiency in creating **Windows applications** using Visual Basic and Visual C++ with event-driven programming, GUI components, and graphics handling.
- To design and implement **database-driven applications** with CRUD operations, ODBC connectivity, and integration of ActiveX/COM components.
- To understand and apply **core Java programming concepts** for secure coding, including object-oriented programming, multithreading, file handling, and exception management.
- To gain hands-on experience in **cybersecurity and network programming**, including malware analysis, encryption, password security, firewall configuration, and safe browsing practices.
- To simulate and implement **network communication and security measures** on Windows and mobile environments, including Winsock programming, secure Wi-Fi setup, and smartphone security configurations.

Course Outcomes:

CO1 Design and develop interactive Windows applications with menus, dialog boxes, controls, graphics, and event handling using Visual Basic and Visual C++.

CO2 Implement database applications with ODBC connectivity, recordsets, data controls, and CRUD operations, and integrate ActiveX/COM components for enhanced functionality.

CO3 Apply object-oriented and Java programming techniques to solve practical problems, including file handling, multithreading, exception handling, and string manipulations.

CO4 Demonstrate practical knowledge of cybersecurity principles, including malware analysis, encryption/decryption, password management, firewall configuration, and secure browsing/email/social media practices.

CO5 Develop and test networked applications for client-server communication, secure Wi-Fi networks, and smartphone security, ensuring robust and secure system operation.

Credits: **02**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in Windows Programming:

1. Develop a Visual Basic Windows application that demonstrates event-driven programming by creating a basic window with menus, accelerators, and message handling, and utilize device contexts to draw simple graphics such as lines, rectangles, and circles.
2. Develop a Visual Basic application showcasing intrinsic controls like text boxes, buttons, and labels, implement event handling for user interactions such as button clicks and menu selections, and incorporate custom controls with programmatic property manipulation.
3. Create a Visual Basic program using modal and modeless dialog boxes, implement a common dialog box for opening and saving files, and demonstrate basic file operations like reading from and writing to files through VB controls.
4. Develop a Visual C++ application that draws graphics such as lines, shapes, and text on a window, implements animation using Picture Clip controls with clipboard operations, and demonstrates printing graphics and text to a printer device context.
5. Develop a Visual Basic program that connects to a database using ODBC, implements CRUD operations on a sample database, and demonstrates recordsets and data control components for displaying and manipulating database records.
6. Develop a Windows Socket application using Winsock for simple client-server communication, incorporate ActiveX, COM, or DCOM components in a Visual Basic project, and implement exception handling and debugging to ensure robust operation and integration.

List of Practical in Java Programming:

1. Analyze sample malware files in a controlled environment using antivirus tools and sandboxing techniques, and prepare a report classifying malware types and their potential impact.
2. Implement basic encryption and decryption techniques using tools or python programming (e.g., Caesar cipher, RSA) and demonstrate the creation and verification of digital signatures for a sample message.
3. Generate secure passwords using a password manager, enable two-step verification on sample online accounts, and document best practices for secure authentication.
4. Configure the Windows or Mac firewall to block/allow specific applications or ports, test network access rules, and analyze the effect on system security and application behavior.
5. Simulate safe browsing practices by configuring browser security settings, analyzing HTTPS certificates, clearing cache, and demonstrating safe email and social media usage including secure posting and privacy settings.
1. Set up a secure Wi-Fi network with WPA2/WPA3 encryption, configure router settings, and implement smartphone security features such as app permissions, secure communication (voice and messages), and device encryption.

Suggested Readings:
https://mrcet.com/pdf/Lab%20Manuals/Lab%20Manual%20Object%20Oriented%20Programming%20through%20JAVA.pdf

Programme: **BCA**		Year: **Third**	Semester: **V**
Subject: BCA			
Course Code: **BCA-5.1**		Course Title: **Python Programming**	
Course Objectives: Provide students with hands-on experience on python programming for solving data science problems.			
Course Outcomes: **CO 1:** The students develops a sound approach to problem solving using a high level programming language. **CO 2:** Use Python data structures – lists, tuples & dictionaries for representing compound data. **CO 3:** The students master the good programming practices like modularity and documentation, and use of named constants. **CO 4:** The student learns the use of object oriented framework using the concept of classes, inheritance, and encapsulation.			
Credits: **04**		Type of Course: **Core**	
Max. Marks: **100**		Min. Passing Marks: **36**	
Block 1	**Basics of Python**		
Unit I	Introduction: History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL (Shell), Running Python Scripts, Python IDLE.		
Unit II	Tokens and Statements: Variables, Constants, Assignment, Multiple Assignment, Keywords, Punctuators, Identifiers, Input-Output, Indentation, Statements, Comments, Single Comment and Multiline Comment.		
Unit III	Data Types, Operators & Expressions: Types – Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Operators precedence, Expressions and order of evaluations Control Flow- if, if-else, if-elif-else, for, while, break, continue, pass.		
Block 2	**Data Structure in Python**		
Unit IV	Data Structures: Stack & Queue, Lists – Operations, Slicing, Methods; Tuples – Operations, Methods , Sets– Operations , Methods, Dictionaries– Operations , Methods, Sequences– Operations, Methods. Comprehensions– Operations, Methods.		
Unit V	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.		
Unit VI	Modules & Packages: Modules, Creating modules, import statement, from. Import statement, name spacing, Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages.		
Block 3	**Oops in Python**		
Unit VII	Object-Oriented Programming OOP in Python: Classes, 'self-variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data hiding.		
Unit VIII	Exception Handling :Error, and Exceptions: Difference between an error and Exception,		

	Handling Exception, try except for block, Raising Exceptions, User Defined Exceptions
Unit IX	Python Libraries: Brief Tour of the Standard Library – Operating System Interface – String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression.
Unit X	GUI Programming and Testing : Multithreading, GUI Programming, Turtle Graphics Testing: Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.
Block 4	**Machine Learning in Python**
Unit XI	Machine Learning Using Python: Machine Learning Basics, Features and Labels, Supervised and Unsupervised Learning.
Unit XII	Regression and Classification in Machine Learning: Simple Linear Regression, Multiple Regression, Data Collection for Machine Learning, Classification – Features and Types.

Suggested Readings:

1. Python Programming: A Modern Approach, VamsiKurama, Pearson
2. Learning Python, Mark Lutz, Orielly
3. Think Python, Allen Downey, Green Tea Press
4. Core Python Programming, W.Chun, Pearson.
5. Introduction to Python, Kenneth A. Lambert, Cengage

Suggested online courses (MOOCs)

1. NOC:Programming, Data Structures and Algorithms using Python, Chennai Mathematical Institute by Prof. Madhavan Mukund
https://nptel.ac.in/courses/106106145
2. NOC:The Joy of Computing using Python, IIT Ropar by Prof. Sudarshan Iyengar
https://nptel.ac.in/courses/106106182
3. Python for Data Science By Prof. Ragunathan Rengasamy, IIT Madras
https://onlinecourses.nptel.ac.in/noc22_cs32/preview

This course can be opted as an elective by the students of following subjects: **B.Sc.(Computer Science), M.Sc. (Statistics) and M.Sc. (Mathematics)**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **BCA**	Year: **Third**	Semester: **V**
Subject: BCA		
Course Code: **BCA-5.2**	Course Title: **Multimedia**	

Course Objectives: Today, Multimedia and web design technology play an essential role in education, agriculture, product launch, science and technology, corporate development and enhanced business opportunities. The increasing variety of hardware and software components in multimedia and website design has escalated the demand for human resources in these fields. This course is designed to inculcate required skills for these activities.

Course Outcomes:

CO1 Visualize scopes of multimedia and understand steps in creation of multimedia applications.

CO2 Understand digital audio, prepare audio required for a multimedia system and Speech synthesis and recognition concept.

CO3 Analyze representation of video, how video work and different video formats.

CO4 Describe different animation techniques and software used for animation.

CO5 Understand various multimedia development and authoring tools.

CO6 Know the different layers of network along with video conferencing technique.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Introduction to Multimedia and Its Components**
Unit 1	**Multimedia Technology:** Meaning & scope of Multimedia; Elements of Multimedia; Creating multimedia applications; Multimedia file & I/O functions; Multimedia data structures; Multimedia file formats; Multimedia Protocols
Unit 2	**Multimedia Audio:** Digital sound; Audio compression & decompression; Companding: ADPCM compression; MPEG audio compression; True Speech; Special effects and Digital Signal Processing: Audio synthesis; FM synthesis: Sound blaster card; Special effect processors on sound cards; Wave table synthesis; MIDI functions; Speech synthesis & Recognition
Unit 3	**Multimedia Video:** Representation of Digital video; Video capture: Frame grabbing; Full motion video; Live video in a window; Video processor; Video compression & decompression; Standards for video compression & decompression; Playback acceleration methods
BLOCK-2	**Multimedia Animation, Authoring Tools and Internet**
Unit 4	**Creating Multimedia Animation:** Icon animation; Bit-map animation; Real-time vs Frame by Frame animation; Object modeling in 3D animation; Motion control in 3D animation; Transparency; Texture. Shadows, Anti-aliasing; Human modeling & Animation; Automatic motion control
Unit 5	**Multimedia Authoring Tools:** Project editor; Topic editor; Hot-spot editor; Developing a multimedia title; Multimedia text authoring systems; Usage of authoring tools
Unit 6	**Multimedia on LANs & Internet:** Multimedia on LAN; Fast modems & Digital networks for multimedia; High speed digital networks; Video conferencing techniques; Multimedia interactive applications on Internet: Future Directions.

Suggested Readings:

1. "Li, Ze-Nian, Mark S. Drew, and Jiangchuan Liu. Fundamentals of multimedia. Upper Saddle River (NJ) Pearson Prentice Hall, 2004.
2. Jeffcoate, Judith. Multimedia in practice: technology and applications. Prentice-Hall, Inc., 1995.
3. Vaughan, Tay. Multimedia: Making it work. Tata McGraw-Hill Education, 2006.
4. Melliar-Smith, Peter Michael, and Louise E. Moser. "Multimedia Networking: Technology, Management and Applications. Hershey, PA Idea Group, 2002.

Suggested online courses (MOOCs)

1. Multimedia processing, IIT Kharagpur by Prof. Somnath Sengupta https://nptel.ac.in/courses/117105083 2. CIT-003: Web Based Technologies and Multimedia Applications By Prof. P. V. Suresh | Indira Gandhi National Open University https://onlinecourses.swayam2.ac.in/nou20_cs05/preview
This course can be opted as an elective by the students of following subjects: N.A.
Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **BCA**	Year: **Third**	Semester: **V**
Subject: BCA		
Course Code: **BCA-5.3**	Course Title: **Soft Computing**	
Course Objectives: Expose students to Neural Network, Fuzzy Logic and Genetic Algorithms, which are the major building blocks of Intelligent Systems.		
Course Outcomes: **CO1** Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human experience. **CO2** Understand how neural networks learn from available examples and generalize to form appropriate rules for inference systems. **CO3** Provide the mathematical background for carrying out the optimization associated with neural network learning. **CO4** Apply genetic algorithms and other random search procedures for finding global optimum of optimization problems.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Artificial Intelligence & Soft Computing**
Unit I	**Introduction of Artificial Intelligence** Definitions, Theoretical background, AI problem domain, General AI techniques, Underlying assumptions, possible goal of AI, Criteria of success. Problem state, state space, search space, State space representation, Production system, control strategy, water jug problem, 8-puzzle problem, Heuristic searching.
Unit II	**Knowledge Representation Model** First order predicate logic, clauses, inference, rule base system, natural deduction and resolution, monotonic reasoning.
Unit III	**Non-Monotonic Reasoning** Uncertainty, Bay's theorem, Bayesian network, dependency network, limitation of probabilistic reasoning, Soft computing definition, soft computing paradigm, applications, Pattern recognition, pattern classification, association and mapping.
Block 2	**Fuzzy Set Theory**
Unit IV	**Introduction of Fuzzy Logic** Uncertainty, Fuzzy set, Crisp vs. fuzzy sets, Membership function, Fuzzy sets and operations, Operations and relations; fuzzy relations, cardinalities, membership functions.
Unit V	**Fuzzy Relations** Fuzzy Cartesian product, fuzzy membership function formulation and parameterization, Fuzzy rules and reasoning, Formulation on fuzzy rules, extension principle and nested fuzzy relations.
Unit VI	**Fuzzy Rule Base System**

	Fuzzy if-then rules, fuzzy inference, Fuzzy inference system, Defuzzification methods, Fuzzy control systems, and Applications of Fuzzy control systems.
Block 3	**Neural Network**
Unit VII	**Introduction of Neural Networks** Limitations of Rule based system, characteristics of neural networks, simple structure of biological neuron and modeling of artificial neuron. Difference between ANN and biological neural networks, artificial neuron models, artificial neural networks terminology, topology of ANN, Characteristics of ANN and its applications.
Unit VIII	**Activation and Synaptic Dynamics** Basic learning laws, Artificial neural network architectures, Basic artificial neural network models, perceptron architecture, Perceptron learning rule, ADLINE architecture, LMS learning rule, Linear classifier, convergence theorem, limitation of perceptron learning, Multi-layer perceptron architecture.
Unit IX	**Pattern Mapping Network** Multilayer feed forward neural network architecture, Generalized delta learning rule, Backpropagation learning algorithm and issues, limitation of Backpropagation learning rule, improvement in BP algorithms, momentum term, conjugate descent, reuse gradient, generalization and approximation, ill posing, Radial basis network.
Block-4	**Genetic Algorithm**
Unit X	**Introduction of Genetic Algorithm** Fundamental and basic concepts, terminology, Applications and advantages, Representation of chromosomes and gens, Population representation, working principle, search space, solution state, global vs local optimization, encoding methods.
Unit XI	**Population Representation** Selection criteria and methods, fitness evaluation function, reproduction, basic genetic operators, Mutation, selection, crossover. Fitness criteria, convergence of GA, combinatorial optimization.
Unit XII	**Problem Solution and Genetic Modeling**, Inheritance operator, crossover operator and its various forms, inversion & deletion, mutation operator, bitwise operator, Generation cycle, Differences & similarities between GA & other traditional method.
Suggested Readings:	

Text Books

1. J.S.R.Jang, C.T.Sun and E.Mizutani, "Neuro-Fuzzy and Soft Computing", PHI, 2004, Pearson Education 2004.
2. S. Rajasekaran and G.A.VijaylakshmiPai.. Neural Networks Fuzzy Logic, and Genetic Algorithms, Prentice Hall of India.
3. Timothy J.Ross, "Fuzzy Logic with Engineering Applications", McGraw-Hill, 1997.
4. Davis E.Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley,N.Y.,1989.
5. S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI, 2003.
6. R.Eberhart, P.Simpson and R.Dobbins, "Computational Intelligence - PC Tools", AP Professional, Boston, 1996.

Suggested online courses (MOOCs)

1. NOC:Introduction to Soft Computing, IIT Kharagpur by Prof. Debasis Samanta https://nptel.ac.in/courses/106105173

This course can be opted as an elective by the students of following subjects: **M.Sc. (Statistics) and M.Sc. (Mathematics)**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: BCA	Year: **Third**	Semester: **V**
Subject: BCA		
Course Code: **BCA-5.4P**	Course Title: **Practical Work**	

Course Objectives:

- To develop proficiency in Python programming fundamentals, including variables, data types, operators, control structures, and input/output operations.
- To design and implement data structures, functions, modules, and object-oriented programs in Python for real-world problem solving.
- To handle exceptions, perform file I/O operations, and create graphical user interfaces (GUI) with multithreading and event-driven programming in Python.
- To apply Python programming for solving machine learning tasks, supervised learning, regression, classification, and evaluating model performance.
- To implement AI and soft computing techniques, including search algorithms, rule-based reasoning, fuzzy logic, neural networks, and genetic algorithms for optimization problems.

Course Outcomes:

CO1 Demonstrate Python programming skills, including control flow, data structures, functions, modules, and OOP principles, to solve computational problems efficiently.
CO2 Develop Python programs with exception handling, file I/O, GUI, and multithreading, ensuring robust and interactive applications.
CO3 Implement machine learning tasks using Python libraries, perform data preprocessing, model training, evaluation, and interpret results accurately.
CO4 Apply artificial intelligence and soft computing techniques using Python to solve search problems, fuzzy logic tasks, neural network classification, and optimization problems.
CO5 Analyze, compare, and optimize solutions obtained from AI algorithms, machine learning models, and soft computing methods for real-world applications.

Credits: **04**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in Python Programming:

1. Write Python programs to demonstrate the use of variables, constants, data types, arithmetic and logical operators. Implement control structures using `if`, `if-else`, `if-elif-else`, `for`, and `while` loops, including the use of `break`, `continue`, and `pass`. Practice input/output operations using the REPL and Python IDLE.
2. Implement Python programs to perform operations on lists, tuples, sets, and dictionaries, including adding, deleting, updating, slicing, and iterating. Demonstrate comprehension techniques to create lists, sets, or dictionaries dynamically.
3. Design Python programs to implement stack and queue data structures using both lists and collections modules. Include operations such as push, pop, enqueue, dequeue, and display, handling overflow and underflow scenarios.
4. Write Python functions demonstrating parameter passing, default arguments, keyword arguments, variable-length arguments, and return values. Create a module with multiple functions, import it in a separate script, and demonstrate function calls using both `import` and `from ... import ...`.
5. Develop Python classes with constructors, methods, and attributes. Demonstrate inheritance, method overriding, and data hiding. Implement a small project, such as a Bank Account system or Student Management System, using OOP principles.
6. Write Python programs that handle exceptions using `try-except`, raise custom exceptions, and perform file handling operations such as reading from and writing to text files. Demonstrate

handling multiple exceptions and ensuring resources are properly closed.

7. Create a Python GUI application using `Tkinter` or `Turtle` graphics. Include interactive components (buttons, labels, entry fields), simple animations, and multithreading for simultaneous tasks. Test the application functionality using unit testing and writing basic test cases.
8. Develop Python programs for supervised learning tasks: implement simple linear regression and multiple regression using sample datasets, and perform classification using features from the dataset. Demonstrate data collection, preprocessing, and model evaluation metrics (accuracy, MSE, etc.).

List of practical in Soft Computing with Python Programming:

1. Implement Python programs to solve state-space search problems such as the 8-puzzle and water jug problem using breadth-first search, depth-first search, and heuristic search techniques. Compare performance and number of steps for each approach.
2. Develop a rule-based system that uses first-order predicate logic to represent facts and rules. Implement inference mechanisms such as forward chaining and backward chaining, and demonstrate reasoning using example queries.
3. Create a Bayesian network for a simple domain (e.g., medical diagnosis or weather prediction). Implement Bayes 'theorem for probabilistic inference and demonstrate dependency relationships and reasoning under uncertainty.
4. Write programs to define fuzzy sets with different membership functions. Perform fuzzy set operations (union, intersection, complement) and demonstrate fuzzy reasoning using if-then rules. Visualize the membership functions graphically.
5. Design a fuzzy rule-based system for a real-world application, such as temperature control or speed control. Implement fuzzification, inference, and defuzzification methods, and demonstrate system output for different input scenarios.
6. Implement a single-layer perceptron in Python to solve a binary classification problem. Demonstrate training using the perceptron learning rule and evaluate convergence, accuracy, and limitations.
7. Design and implement a multilayer feedforward neural network using Python (NumPy, TensorFlow, or PyTorch) for a classification task. Apply backpropagation algorithm, experiment with learning rate and momentum, and analyze performance on training and test data.
8. Implement a genetic algorithm to solve a combinatorial optimization problem such as travelling salesman problem or function optimization. Demonstrate chromosome representation, selection, crossover, mutation, and fitness evaluation, and analyze convergence toward optimal solutions.

Programme: **BCA**	Year: **Third**	Semester: **V**
Subject: BCA		
Course Code: **BCA-EA**	Course Title: **Web Technology**	
Course Objectives: This course intended to also expose students to the basic tools and technologies used in the development of a web application for the World Wide Web. This includes – Internet technologies, HTML, XML, JavaScript, and JSP.		
Course Outcomes: **CO1** Implement interactive web page(s) using HTML, CSS and JavaScript. **CO2** Design a responsive web site using HTML5 and CSS **CO3** Build Dynamic web site using server side PHP Programming and Database connectivity. **CO4** Describe and differentiate different Web Extensions and Web Services. **CO5** Understand development of web application.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	History of the Internet and World Wide Web -III ML 4 protocols - RCM, SMTP, POP), MIME, IMAP. Introduction to JAVA Scripts - Object Based Scripting for the web, Structures - Functions - Arrays - Objects.	
Block 2	Introduction - Object refers, Collectors all and Children. Dynamic style, Dynamic position, frames. navigator, Event Model - On check - On load - - Form process - Event Bubblers- filters -Transport with the Filter - Creating Images Adding shadows - Creating Gradients - Creating Motion with Bar-Data Binding - Simple Data Binding - Moving with a record set - Sorting table data, binding of an image and table.	
Block 3	database, Relational Database model - Overview, SQL - ASP - Working of ASP - Objects - File System Objects - Session tracking and cookies - ADO - Access a Database from ASP - Server side Active-X Components - Web Resources - XMIL - Structure in Data Name spaces - D7D- Vocabularies - DOM methods.	
Block 4	Introduction, Servlet, Overview Architecture - Dandling II P Request - Go and post request - redirecting request multi-tier applications - ISP - Overviews - Objects - scripting - Standard Actions - Directives. Brief survey of Web 2.0 technologies, introduction to Semantic web and other current technologies.	
Suggested Readings: 1. Deitel, Deitel, Goldberg, "Internet & World Wide Web How to Program", Third Edition, Pearson Education, 2006. 2. Raj Kamal, “Internet and Web Technologies”, Tata McGraw-Hill.		
This course can be opted as an elective by the students of following subjects: **B.Sc.(Computer Science), MCA**		
Suggested equivalent online courses (MOOCs) for credit transfer: N.A		

Programme: **BCA**	Year: **Third**	Semester: **V**
Subject: BCA		
Course Code: **BCA-EB**	Course Title: **Client Server Technology**	
Course Objectives: This course gives fundamental principles of Client-Server technology. The course revolves around the knowledge of ASP.NET framework and its related components like form and controls, state management, and configuration. It discusses ASP.NET web services, HTML and JavaScript, DHTML, AJAX, and a small application.		
Course Outcomes: **CO1:** Design web applications using ASP.NET. **CO2:** Use ASP.NET controls in the web applications. **CO3:** Create database driven ASP.NET web applications. **CO4:** Incorporate session state, application state and cookies in the web application.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Introduction to Client-Server Computing**
Unit 1	Introduction to Client-Server Computing: Introduction to Client-Server Architecture, Client-Server computing and its uses, historical development, downsizing and client server computing, mainframe computing, client-server technology and heterogeneous computing, advantages of client server computing.
Unit 2	Distributed Computing: Distributed Computing, File Server versus Client/Server Database, Computing platforms, Microprocessor integration and client server computing, implementations and scalability.
Unit 3	Designing Client-Server Applications: Fundamentals of client server design, division of labor, Transition to client-server programming; Interaction of client and server communication Techniques and protocols, implementing client server applications.
Block 2	**Introduction to ASP.NET**
Unit 4	Introduction to .NET Framework: Introduction, The Origin of .Net Technology, Common Language Runtime (CLR), Common Type System (CTS), Common Language Specification (CLS), Microsoft Intermediate Language (MSIL), Just-In –Time Compilation, Framework Base Classes.
Unit 5	Traditional ASP Basics: Introduction to ASP, How ASP Works, ASP Objects, Installing IIS on Windows 7 & Windows 8, Sample Programs, Importance's of Form tag and how it works.
Unit 6	ASP.NET Introduction & Controls: ASP.NET Introduction, First ASP.NET Application, Auto Postback Property, Event Handler, Parameters, Dynamically intializing Controls, IsPostBack property of Page class, ListControls, Comparison between HtmlControls and WebControls, Control Properties and Methods, FileUpload Control
Block 3	**Working with Forms and Controls**
Unit 7	Working with Forms and Controls: Life Cycle of ASP.NET Page, Creating an ASP.NET Web Application Project, Creating Web Forms, Using Server Controls, Using Code-Behind Pages, Web Server Controls, Using Validation controls usage of skins and themes.
Unit 8	ADO.Net: Introduction to ADO.NET, .NET Framework data providers, Data Binding, Connecting to the Database, Accessing Data with DataSets, Displaying a DataSet in a List-Bound Control, Using Multiple Tables, Accessing Data with DataReaders, Disconnected operations with Data tables and Data sets, Connection pooling, Working with LINQ.
Unit 9	ASP.NET State Management: Application and Session Variables, Cookies, Storing Session Variables in a Database, Cleaning the session state, Types of Assemblies, Private vs. Shared assemblies, Creating and placing strongly named assemblies.

Unit 10	Configuration: Windows configuration, .net configuration, cashing, Types of Caching, SQL Cache Invalidation
Block 4	**Client Side and Server Side Login Services**
Unit 11	HTML & JavaScript: Understanding HTML Form Tag and elements within it, Javascript using Sample Programs, Working with CSS, Use Themes to Customize a Site,Web based security, ASP.NET authentication service, managing user,asp.net login controls, authorizing users.
Unit 12	ASP.Net Web Services: Introduction to web services, creating web services, invoking web services,
Unit 13	AJAX: Introduction to AJAX, AJAX.NET, Script Manager, Update Panel, Update Progress, Timer, AJAX Control Toolkit, server side support for AJAX, AJAX client support.
Unit 14	Developing a small application using ASP.NET for any case study.
Suggested equivalent online courses (MOOCs) for credit transfer: N.A	

Programme: **BCA**	Year: **Third**	Semester: **VI**
Subject: BCA		
Course Code: **BCA-6.1**	Course Title: **Information and Network Security**	
Course Objectives: This course aims to provide a basic understanding of the existing algorithms used to protect users online and understand some of the design choices behind these algorithms. The course offers a workable knowledge of the mathematics used in cryptology. The course emphasizes giving a basic understanding of previous attacks on cryptosystems to prevent future attacks.		
Course Outcomes: **CO1** Identify information security goals, classical encryption techniques and acquire fundamental knowledge on the concepts of finite fields and number theory. **CO2** Understand, compare and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication **CO3** Apply the knowledge of cryptographic checksums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes **CO4** Apply different digital signature algorithms to achieve authentication and create secure applications **CO5** Apply network security basics, analyze different attacks on networks and evaluate the performance of firewalls and security protocols like SSL, IPSec, and PGP. **CO6** Apply the knowledge of cryptographic utilities and authentication mechanisms to design secure applications.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Information security and Symmetric Ciphers**
Unit 1	**Introduction:** History, what is Information Security; Characteristics of Information; Information Security Model; Components of an Information Security; Aspects of Information security: Security attacks, Security Mechanism, and Security Services (X.800), Model for Network Security.
Unit 2	**Classical Encryption Techniques: Historical** background, symmetric cipher model, Substitution techniques, Transposition techniques, steganography.
Unit 3	**Block ciphers and DES:** Block cipher principles, Data encryption standard, strength of DES, differential and cryptanalysis, block cipher design principles, block cipher mode of operation.
Unit 4	**Confidentiality Using Symmetric Ciphers:** Placement of encryption function, traffic confidentiality, key distribution, random number generation.
Block 2	**Public key Encryption and Hash Functions**
Unit 5	**Introduction to Number Theory:** Prime numbers, Fermat's and Euler's theorem, discrete logarithm
Unit 6	**Public Key Cryptography:** Public-Key Cryptography Principles, RSA, Key Management: Diffi-Hellman key exchange.
Unit 7	**Message Authentication and Hash Functions:** Authentication requirements, Authentication Functions, Message Authentication codes, Hash Functions, SHA-1, MD5.
Unit 8	**Digital Signatures:** Digital signatures, Authentication protocols, Digital Signature standard
Block 3	**Network Security Applications**
Unit 9	**Authentication Applications:** Kerberos Motivation, X.509 authentication service
Unit 10	**Electronic Mail Security:** PGP: PGP Notation, PGP Operational Description, S/MIME
Unit 11	**IP Security: IP** Security Overview, IP Security Architecture, Authentication Header

Unit 12	**Web Security:** Web Security Threats, Web Traffic Security Approaches, Overview of Secure Socket Layer and Transport Layer Security, Overview of Secure Electronic Transaction
Block 4	**Intruders and Viruses**
Unit 13	**Intruders:** Intruders, Intrusion Techniques, Password Protection, Password Selection Strategies, Intrusion Detection,
Unit 14	**Malicious Programs:** Malicious Programs, Nature of Viruses, Types of Viruses, Macro Viruses, Antivirus Approaches
Unit 15	**Firewall**: Firewall Characteristics, Types of Firewalls, Firewall Configuration

Suggested Readings:

1. Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman & Hall/CRC.
2. B. A. Forouzan, "Cryptography & Network Security", Tata Mc Graw Hill.
3. W. Stallings, "Cryptography and Network Security", Pearson Education.

Suggested online courses (MOOCs)

1. NOC:Cryptography And Network Security, IIT Kharagpur by Prof. Sourav Mukhopadhyay https://nptel.ac.in/courses/106105162
2. Cryptography and Network Security, IIT Kharagpur by Dr. Debdeep Mukhopadhyay https://nptel.ac.in/courses/106105031

This course can be opted as an elective by the students of following subjects: **MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **BCA**	Year: **Third**	Semester: **VI**
Subject: Computer Science		
Course Code: **BCA-6.2**	Course Title: **Computer Graphics**	
Course Objectives: The primary role of computer graphics is to render the digital content (0's and 1's) in a human-comprehensible form on the computer screen. This course introduces various object representation techniques along with 2D and 3D transformation, clipping, splines, objects modeling, colour modeling, lighting, textures and visible surface detection.		
Course Outcomes: **CO1** Demonstrate an understanding of contemporary graphics hardware. **CO2** Draw graphics using line & polygon and ability to perform operations on computer graphics. **CO3** Understand and demonstrate geometrical transformations, Segment, Windowing and Clipping, Interaction. **CO4** Demonstrate Hidden Surfaces & Lines; Light, Colour & Shading; Curves and Fractals		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Raster Graphics and Clipping**
Unit 1	Unit 1: Introduction to Computer Graphics: What is Computer Graphics?, Application of Computer Graphics, Presentation Graphics, Painting and Drawing, Photo Editing, Scientific Visualization, Image Processing, Digital Art, Education, training, Entertainment and CAD Simulation, Animation and Games, Graphics Hardware, Input and Output Devices, Touch Panel, Light Pens, Graphic Tablets, Plotters, Film Recorders, Display Devices, Refreshing Display Devices: Raster-Scan, Random-Scan, Plasma Panel and LCD panels
Unit 2	Unit 2: Graphics Primitives: Points and Lines, Line-drawing Algorithms: DDA Algorithm, Bresenham's line Algorithm, Circle-generating Algorithm: Properties of Circles, Midpoint Circle of Algorithm, Polygon Filling Algorithm: Scan-Line
Unit 3	Unit 3: 2-D Viewing and Clipping: Point Clipping, Line Clipping: Cohen-Sutherland Line Clippings, Cyrus-Beck Line Clipping Algorithm, Polygon Clipping: Sutherland Hodgman Algorithm, Windowing Transformation
Block 2	**Transformations**
Unit 4	Unit 4: 2-D and 3-D Transformations: Basic Transformations: Translation, Rotation, Scaling, Shear, Composite Transformations: Rotations about a point, Reflection about a line, Homogeneous Coordinate Systems, 3-D Transformations
Unit 5	Unit 5: Viewing Transformation: Projections: Parallel Projection, Orthographic & Oblique Projections, Isometric Projections, Perspective Projections
Block 3	**Modeling & Rendering**
Unit 6	Unit 6: Curves and Surfaces: Polygon Representation Methods: Polygon Surfaces, Polygon Tables, Plane Equations, Polygon Meshes, Bezier Curves and Surfaces: Bezier Curves, Properties of Bezier Curves, Bezier Surfaces, Surface of Revolution
Unit 7	Unit 7: Visible – Surface Detection: Depth Buffer Method, Scan-Line Method, Area-Subdivision Method
Unit 8	Unit 8: Polygon Rendering and Ray Tracing Methods: Illumination Model: Ambient Reflection, Diffuse Reflection, Specular Reflection, Shading: Gouraud Shading, Phong Shading, Ray Tracing: Basic Ray-Tracing Algorithm

Suggested Readings:

2. J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics - Principles and Practice, Second Edition in C, Pearson Education, 2003.
3. D. Hearn and M. Pauline Baker, Computer Graphics (C Version), Pearson Education, 2nd Edition, 2004.
4. Edward Angel, Interactive Computer Graphics A Top-Down Approach with OpenGL 5th Edition,

Addison-Wesley, 2008.
5. Prabat K Andleigh and KiranThakrar, "Multimedia Systems and Design", PHI, 2003.

Suggested online courses (MOOCs)

1. Computer Graphics, IIT Madras by Prof. Sukhendu Das
   https://nptel.ac.in/courses/106106090
2. Introduction to Computer Graphics, IIT Delhi by Prof. Prem K Kalra
   https://nptel.ac.in/courses/106102065
3. NOC:Computer Graphics, IIT Guwahati by Prof. Samit Bhattacharya
   https://nptel.ac.in/courses/106103224

This course can be opted as an elective by the students of following subjects: This course can be opted as an elective by the students of following subjects: **B.Sc. (Computer Science) and BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **BCA**	Year: **Third**	Semester: **VI**
Subject: **BCA**		
Course Code: **BCA-6.3P**	Course Title: **Project with viva voce**	
Course Objectives: ➢ To facilitate the learner to independently formulate and solve a social, philosophical, commercial, or technological problem and present the results in written and oral form. ➢ To render learners to real-life problems. ➢ To provide opportunities for learners to interact with people and present them confidently.		
Course Outcomes: **CO1** Investigate and evaluate a research topic relevant to environment and society. **CO2** Learn systematic discovery and critical review of appropriate and relevant information sources. **CO3** Apply qualitative and/or quantitative evaluation processes to original data. **CO4** Communicate research concepts and contexts clearly and effectively both in writing and orally		
Credits: **08**	Type of Course: **Application**	
Max. Marks: **100**	Min. Passing Marks:	

Programme: **BCA**	Year: **Third**	Semester: **VI**
Subject: BCA		
Course Code: **BCA-EC**	Course Title: **Computer Architecture**	
Course Objectives: The course aims to impart understanding of fundamental and advanced concepts of parallel computing and design architecture. It illustrates concepts of memory and input-output subsystems, pipelining and vector processing, microprocessor algorithms and systems and control mechanisms.		
Course Outcomes: **CO1** Familiarizes the students with basics of computer hardware and how software interacts with computer hardware. **CO2** Introduces how computers represent and manipulate data, computer arithmetic and conversion between different number systems. **CO3** Introduces how Boolean algebra is related to designing computer logic, through simple combinational and sequential logic circuits. **CO4** Introduces basics of Instruction Set Architecture (ISA). **CO5** Familiarize students with a simple computer with hardware design including data format, instruction format, instruction set, addressing modes, bus structure, input/output, memory, Arithmetic/Logic unit, control unit, and data, instruction and address flow. **CO6** Design combinational and sequential logic circuits, flip-flops, counters, shift registers, adders, substractor, multiplexer, demultiplexer, Arithmetic/Logic unit. **CO7 I**ntroduces concept of memory unit and input/output architecture.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Processor Basics**
Unit 1	**CPU organization:** Fundamentals, additional features
Unit 2	**Data representation:** Basic formats, fixed point numbers, floating point numbers
Unit 3	**Instruction sets:** Instruction formats, instruction types, programming considerations
Block 2	**Data path Design**
Unit 4	Fixed point arithmetic Addition and subtraction, multiplication and division
Unit 5	**Arithmetic Logic Unit:** Combinational ALUs, sequential ALUs
Unit 6	**Advanced Topics:** Floating point arithmetic, pipeline processing
Block 3	**Control Design**
Unit 7	**Basic concepts:** Introduction, hardwired control, design examples
Unit 8	**Micro programmed control:** Basic concepts, multiplier control unit, CPU control unit
Unit 9	**Pipeline control:** Instruction pipelines, pipeline performance, super scalar processing
Block 4	**Memory Organization**
Unit 10	**I/O and System:** Control Programmed IO, DMA and Interrupts, 10 processors
Unit 11	Parallel processing: Processor-level parallelism, multiprocessor

Suggested Readings:

1. Computer Organization & Architecture - Designing for Performance by William Stallings, Eighth Edition, Pearson, 2010
2. Computer Architecture: A Quantitative Approach by John L. Hennessy and David A. Patterson, Fourth Edition, Morgan Kaufmann Publishers
3. Computer System Architecture by M. Morris Mano, Third Edition, Pearson Education Inc

Suggested online courses (MOOCs)

1. Computer Architecture, IIT Delhi by Prof. Anshul Kumar https://nptel.ac.in/courses/106102062
2. NOC:Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof.

Kamalika Datta https://nptel.ac.in/courses/106105163 3. NOC:Computer Architecture, IIT Delhi by Prof. Smruti R.Sarangi https://nptel.ac.in/courses/106102157 4. NOC:Computer Architecture(Course sponsored by Aricent), IIT Madras by Prof.Madhu Mutyam https://nptel.ac.in/courses/106106134
This course can be opted as an elective by the students of following subjects: **M.Sc.(Computer Science), MCA**
Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **BCA**	Year: **Third**	Semester: **VI**
Subject: BCA		
Course Code: **BCA-ED**	Course Title: **Microprocessor and its applications**	
Course Objectives: This course provides architecture and organization of microprocessor along with instruction set format. It discusses modes and functional block diagram of 8085 AND 8086 along with pins and their functions; describe memory and addressing modes; explains use different types of instructions, directives and interrupts; illustrates assembly language programs using various programming tools.		
Course Outcomes: **CO1** Apply basic binary math operations using the microprocessor. **CO2** Demonstrate programming using various addressing modes and data transfer instructions of the target microprocessor and microcontroller. **CO3** Compare different Microprocessors (8085 & 8086) and Microcontroller to meet specified performance requirements. **CO4** Analyze and use assembly language programs to solve real-world control problems.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block-1	**Introduction to Microprocessor**
Unit 1	**Introduction of Microcomputer System:** CPU, I/O devices, clock, memory, bussed architecture, tristate logic, address bus, data bus and control bus.
Unit 2	**Semiconductor Memories :** Development of semiconductor memory, internal structure and decoding, memory read and write timing diagrams, MROM, ROM, EPROM, EEPROM, DRAM,
Unit 3	**Architecture of 8-bit Microprocessor:** Intel 8085A microprocessor, Pin description and internal architecture.
Unit 4	**Operation and Control of Microprocessor:** Timing and control unit, op-code fetch machine cycle, Memory read/write machine cycles, I/O read/write machine cycles, interrupt acknowledge machine cycle, state- transition diagram.
Block-2	**Operations, Instruction Set and Assembly Language Programming**
Unit 5	**Instruction Set:** Addressing modes; Data transfer, arithmetic, logical, branch, stack and machine control groups of instruction set, macro RTL and micro RTL flow chart of few typical instructions; Unspecified flags and instructions.
Unit 6	**Assembly Language Programming:** Assembler directives, simple examples; Subroutines, parameter passing to subroutines.
Block-3	**Interface, Interrupt and Programmable Interface**
Unit 7	**Interfacing:** Interfacing of memory chips, address allocation technique and decoding; Interfacing of I/O devices, LEDs and toggle-switches as examples, memory mapped and isolated I/O structure; Input/Output techniques: CPU initiated unconditional and conditional I/O transfer, device initiated interrupt I/O transfer.
Unit 8	**Interrupts:** Interrupt structure of 8085A microprocessor, processing of vectored and non- vectored interrupts, latency time and response time; Handling multiple interrupts
Unit 9	**Programmable Peripheral Interface:** Intel 8255, pin configuration, internal

	structure of a port bit, modes of operation, bit SET/RESET feature, programming; ADC and DAC chips and their interfacing.
Block-4	**Timer, Controllers and Applications**
Unit 10	**Programmable Interval Timer:** Intel 8253, pin configuration, internal block diagram of counter and modes of operation, counter read methods, programming, READ-BACK command of Intel 8254.
Unit 11	**Programmable Interrupt Controller 8253/8254:** Pin configuration, Timer or counter, Internal structure, Interfacing with system, Mode (0,1,2,3,4,5), Reading timer, Read back command feature.
Unit 12	**Programmable Interrupt Controller 8259A:** Priority interrupt structure, Intel 8259, Pin configuration, Functional Block Diagram, Interrupt sequence, Initiallization control words ICW1, ICW2, ICW3, ICW4, Operation Control Words(OCWs), Fully nested mode, EOI mode, Poll command, Reading status registers, Special fully nested mode, Cascade mode.
Unit 13	**Application of Microprocessor:** Various applications of Intel 8085 Microprocessor, Microprocessor based stepper motor control system using 8085, traffic light controller using 8085, ADC interface using 8085, DAC interface using 8085

Suggested Readings:

1. 'Fundamentals of Microprocessors and microcontrollers' by B. Ram, Eighth Revised Edition, Dhanpat Rai Publications.
2. Fundamentals of Microprocessors and Microcontrollers, B. Ram, Dhanpat Rai Publications.
3. Microprocessors and Microcontrollers, S K Mandal. WBUT Series by TMH

Suggested online courses (MOOCs)

1. NOC:Microprocessors And Microcontrollers, IIT Kharagpur by Prof. Santanu Chattopadhyay https://nptel.ac.in/courses/108105102

This course can be opted as an elective by the students of following subjects: **M.Sc.(Computer Science), MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A