BENGALURU NORTH UNIVERSITY
KOLAR - 563103

National Education Policy - 2020
(Semester Scheme)

Model Curriculum Structure for Bachelor of Computer
Applications (BCA) Programme
(Basic and Honours degree)

Model Syllabus for 1st to 4th Semesters
and
Open Elective Courses in Computer Science.

Revised w.e.f.

Academic Year 2021 - 2022 and onwards

Preamble

Computer Application (CA) has been evolving as an important branch of science and technology in last two decade and it has carved out a space for itself like computer science and engineering. Computer application spans theory and more application and it requires thinking both in abstract terms and in concrete terms.

The ever -evolving discipline of computer application has strong connections toother disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers and its applications, but finding a solution requires both computer science expertise and knowledge of the particular application domain.
Computer science has a wide range of specialties. These include Computer Architecture, Software Systems, Graphics, Artificial Intelligence, Mathematical and Statistical Analysis, Data Science, Computational Science, and Software Engineering.

Universities and other HEIs introduced programmes of computer application. Information Technology is growing rapidly. Increasing applications of computers in almost all areas of human endeavour has led to vibrant industries with concurrent rapid change in technology. Unlike other basic disciplines, developing core competency in this discipline that can be reasonably stable becomes a challenge.

In India, it was initially introduced at the Master (postgraduate) level as MCA and M.Tech. Later on, engineering programmes such as B.Tech and B.E in Computer Science & Engineering and in Information Technology were introduced in various engineering College/Institutions to cater to the growing demand for trained engineering manpower in IT industries. Parallelly, BCA, BSc and MSc programmes with specialisation in Computer Science were introduced to train manpower in this highly demanding area.

BCA and BCA (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career.

BCA and BCA (Hons) aims at laying a strong foundation of computer application at an early stage of the career. There are several employment opportunities and after successful completion of BCA, graduating students can fetch employment directly in companies as programmer, Web Developer, Software Engineer, Network Administrator,Data Scientist, or AI/ML personnel.

The Program outcomes in BCA are aimed at allowing flexibility and innovation in design and development of course content, in method of imparting training, in teaching learning process and in assessment procedures of the learning outcomes. The emphasis in BCA courses, in outcome-based curriculum framework, help students learn solvingproblems, accomplishing IT tasks, and expressing creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages.

All students must, therefore, have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn. More importantly, students will learn to adapt to changes in programming languages and learn new languages as they are developed.

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

- To develop an understanding and knowledge of the basic theory of Computer Science and Information Technology with good foundation on theory, systems and applications such as algorithms, data structures, data handling, data communication and computation
- To develop the ability to use this knowledge to analyse new situations in the application domain
- To acquire necessary and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit to the above-mentioned purpose.
- The ability to synthesize the acquired knowledge, understanding and experience for a better and improved comprehension of the real-life problems
- To learn skills and tools like mathematics, statistics and electronics to find the solution, interpret the results and make predictions for the future developments
- To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate

The objectives of the Programme are:

1. The primary objective of this program is to provide a foundation of computing principles and business practices for effectively using/managing information systems and enterprise software

2. It helps students analyze the requirements for system development and exposes students to business software and information systems

3. This course provides students with options to specialize in legacy application software, system software or mobile applications

4. To produce outstanding IT professionals who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves

5. To provide opportunity for the study of modern methods of information processing and its applications.

6. To develop among students the programming techniques and the problem-solving skills through programming

7. To prepare students who wish to go on to further studies in computer science and related subjects.

8. To acquaint students to Work effectively with a range of current, standard, Office Productivity software applications

Program Outcomes: **BCA** (3 Years) **Degree**

1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity

2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.

3. **Design and Development of Solutions:** Ability to design and development of algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.

4. **Programming a computer:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.

5. **Application Systems Knowledge**: Possessing a sound knowledge on computer application software and ability to design and develop app for applicative problems.

6. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tool or technique for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.

7. **Communication:** Must have a reasonably good communication knowledge both in oral and writing.

8. **Project Management:** Practicing of existing projects and becoming independent to launch own project by identifying a gap in solutions.

9. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrality in a working environment and also have concern on societal impacts due to computer-based solutions for problems.

10. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.

11. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on Computer Science.

Additional Program Outcomes: **BCA Degree** (Hons)

The Bachelor of Computer Application (BCA (Hons)) program enables students to attain following additional attributes besides the afore-mentioned attributes, by the time of graduation:

1. Apply standard Software Engineering practices and strategies in real -time software project development

2. Design and develop computer programs/computer -based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics.

3. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems

4. The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.

5. The ability to work independently on a substantial software project and as an effective team member.

Model Curriculum for BCA

Sem	Core Courses	Hour / Week		DS Elective Courses	Hous/ Week
		Theory	Lab		
1	i. Fundamentals of Computers ii. Programming in C iii. Mathematical Foundation/ Accountancy iv. LAB: Information Technology v. LAB: C Programming	3 3 3 	 4 4		
2	i. Discrete Mathematical Structures ii. Data Structures using C iii. Object Oriented Concepts using JAVA iv. LAB: Data Structure v. LAB: JAVA Lab	3 3 3 	 4 4		
3	i. Data Base Management Systems ii. C# and DOT NET Framework iii. Computer Communication and Networks iv. LAB: DBMS v. LAB: C# and DOT NET Framework	3 3 3 	 4 4		
4	i. Python Programming ii. Computer Multimedia and Animation iii. Operating Systems Concepts iv. LAB: Multimedia and Animation v. LAB: Python programming	3 3 3 	 4 4		
5	i. Internet Technologies ii. Statistical Computing and R Programming iii. Software Engineering iv. LAB: R Programming v. LAB: JAVA Script, HTML and CSS vi. Vocational 1	3 3 3 3	 4 4	(a) Cyber Law and Cyber Security (b) Cloud Computing (c) Business Intelligence	3 3 3
6	i. Artificial Intelligence and Applications ii. PHP and MySQL iii. LAB: PHP and MySQL iv. PROJECT: v. Vocational 2	3 3 3	 4 12	(a) Fundamentals of Data Science (b) Mobile Application Development (c) Embedded Systems	3 3 3
7	i. Analysis and Design of Algorithms ii. Data Mining and Knowledge Management iii. LAB: Algorithms iv. LAB: Data Mining and Knowledge Management v. Vocational 3	3 3 	 4 4	(a) Data Compression (b) IoT (c) Data Analytics	3 3 3
8	i. Automata Theory and Compiler Design ii. Cryptography and Network Security iii. Compiler Lab iv. LAB: Project v. Vocational 4	3 3 3	 4 12	(a) Open-Source Programming (b) Storage Area Networks (c) Pattern Recognition (a) Machine Learning	3 3 3 3

TABLE I: COURSE STRUCTURE FOR BCA.

Semester	Course Code	Title of the Paper	Credit	Total Credit of OE, Languages, CAE, Voc, AECC, SEC	Total Credit
I	CAC01	Fundamentals of Computers	3	13	26
	CAC02	Programming in C	3		
	CAC03(a)/(b)	Mathematical Foundation/ Accountancy	3		
	CAC01P	LAB: Information Technology	2		
	CAC02P	LAB: C Programming	2		
II	CAC04	Data Structures using C	3	13	26
	CAC05	Object Oriented Concepts using JAVA	3		
	CAC06	Discrete Mathematical Structures	3		
	CAC04 P	LAB: Data Structure	2		
	CAC05 P	LAB: JAVA	2		
III	CAC07	Data Base Management Systems	3	13	26
	CAC08	C# and DOT NET Framework	3		
	CAC09	Computer Communication and Networks	3		
	CAC07P	LAB: DBMS	2		
	CAC08P	LAB: C# and DOT NET Framework	2		
IV	CAC10	Python Programming	3	13	26
	CAC11	Computer Multimedia and Animation	3		
	CAC12	Operating System Concepts	3		
	CAC10P	LAB: Python programming	2		
	CAC11P	LAB: Multimedia and Animation	2		
V	CAC13	Internet Technologies	3	10	23
	CAC14	Statistical Computing and R Programming	3		
	CAC15	Software Engineering	3		
	CAC13P	LAB: JAVA Script, HTML and CSS	2		
	CAC14P	LAB: R Programming	2		
VI	CAC16	PHP and MySQL	3	10	23
	CAC17	Artificial Intelligence and Applications	3		
	CAC16P	LAB: PHP and MySQL	2		
	CA-P1	Project Work	5		
VII	CAC18	Analysis and Design of Algorithms	3	11	21
	CAC19	Data Mining and Knowledge Management	3		
	CAC18P	LAB: Algorithms	2		
	CAC19P	LAB: Data Mining	2		
	CAI01	Internship	2		
VIII	CAC20	Automata Theory and Compiler Design	3	6	20
	CAC21	Cryptography and Network Security	3		
	CAC20P	LAB: Compiler Lab	2		
	CAP02	Project Work	6		

8

TABLE II: CS COURSE DETAILS FOR BCA

Course-Type	Course Code as referred above	Compulsory/ Elective	List of compulsory courses and list of option of elective courses. (A suggestive list)
CA	CAC01, CAC02, CAC03(a)/(b), CAC04, CAC05, CAC06, CAC07, CAC08, CAC09, CAC10, CAC11, CAC12, CAC13, CAC14, CAC15, CAC16, CAC17, CAC18, CAC19, CAC20, CAC21	Compulsory	As Mentioned in Table I
CA E	CAE-1A	Elective	Cyber Law and Cyber Security OR Business Intelligence OR Fundamentals of Data Science
	CAE-2A	Elective	Fundamentals of Data Science OR Mobile Application Development OR Embedded Systems
	CAE-3A	Elective	Data Compression OR Internet of Things (IoT) OR Data Analytics
	CAE-4A	Elective	Open-source Programming OR Storage Area Networks OR Pattern Recognition OR Machine Learning
Vocational	Vocational -1	Elective	DTP, CAD and Multimedia OR Hardware and Server Maintenance OR Web Content Management Systems OR Computer Networking OR Health Care Technologies OR Digital Marketing OR Office Automation
	Vocational -2	Elective	
	Vocational -3	Elective	
	Vocational -4	Elective	
SEC	SEC 1	Compulsory	Health & Wellness/ Social & Emotional Learning
	SEC 2	Compulsory	Sports/NCC/NSS etc
	SEC 3	Compulsory	Ethics & Self Awareness
	SEC 4	Compulsory	Professional Communication
AECC	AECC1	Compulsory	Environmental Studies
	AECC2	Compulsory	Constitution of India
Language 1	L1-1, L1-2, L1-3, L1-4	Compulsory	Kannada/Functional Kannada
Language 2	L2-1, L2-2, L2-3, L4-4	Elective	English/Hindi/French/ Additional English/ etc.

Model Course Content for BCA, Semesters I and II

Semester: I

Course Code: CAC01	**Course Title:** Fundamentals of Computers
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03

Course Outcomes (COs):

- Introduction to computers, classification of computers, anatomy of computer, constituents and architecture, microcontrollers

- Operating systems, functions of operating systems, classification of operating systems, kernel, shell, basics of Unix, shell programming, booting

- Databases, why databases are used, users, SQL, data types in SQL, introduction of queries - select, alter, update, delete, truncate, using where, and or in not in

- Internet basics, features, applications, services, internet service providers, domain name system, browsing, email, searching

- Web Programming basics, introduction of HTML and CSS programming

- Introduction of computers, classification of computers, anatomy of computer, constituents and architecture, microcontrollers.

Course Content

Content	Hours
Unit – 1	
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.	10

Unit-2	
Introduction to computers: Characteristics of computers, Classification of Digital Computer Systems: Microcomputers, Minicomputers, Mainframes, Super computers. Anatomy of Computer: Introduction, Functions & Components of a Computer, Central Processing Unit, Microprocessor, Storage units, Input and output Devices. How CPU and memory works. Program execution with illustrative examples. Introduction to microcontrollers.	10
Unit-3	
Operating System Fundamentals: Operating Systems: Introduction, Functions of an operating System, Classification of Operating Systems, System programs, Application programs, Utilities, The Unix Operating System, Basic Unix commands, Microkernel Based Operating System, Booting.	10
Unit-4	
Introduction to Database Management Systems: Database, DBMS, Why Database -File system vs DBMS, Database applications, Database users, Introduction to SQL, Data types, Classification of SQL-DDL with constraints, DML, DCL, TCL	6
Unit-5	
Internet Basics: Introduction, Features of Internet, Internet application, Services of Internet, Logical and physical addresses, Internet Service Providers, Domain Name System. **Web Basics:** Introduction to web, web browsers, http/https, URL, HTML5, CSS	6

Text Books:

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC,

Reference:

1. J. Glenn Brook shear," Computer Science: An Overview", Addision-Wesley, Twelfth Edition,
2. R.G. Dromey, "How to solve it by Computer", PHI,

Course Code: CAC01P	**Course Title: Information Technology Lab**
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 04

Part A: Hardware

1. Identification of the peripherals of a computer, components in a CPU and their functions.

2. Assembling and disassembling the system hardware components of personal computer.

3. Basic Computer Hardware Trouble shooting.

4. LAN and WiFi Basics.

5. Operating System Installation – Windows OS, UNIX/LINUX, Dual Booting.

6. Installation and Uninstallation of Software – Office Tools, Utility Software (like Anti-Virus, System Maintenance tools); Application Software - Like Photo/Image Editors, Audio Recorders/Editors, Video Editors …); Freeware, Shareware, Payware and Trialware; Internet Browsers, Programming IDEs,

7. System Configuration – BIOS Settings, Registry Editor, MS Config, Task Manager, System Maintenance, Third-party System Maintenance Tools (Similar to CCleaner and Jv16 PowerTools …)

Part B: Software

1. Activities using Word Processor Software

2. Activities using Spreadsheets Software

3. Activities using Presentation Software

4. Activities involving Multimedia Editing (Images, Video, Audio …)

5. Tasks involving Internet Browsing

6. Flow charts: Installation and using of flowgarithms software for different arithmetic tasks like sum, average, product, difference, quotient and remainder of given numbers, calculate area of Shapes (Square, Rectangle, Circle and Triangle), arrays and recursion.

NOTE: In addition to the ones listed above, universities can include other activities so as for the student to become proficient in using personal computers for multiple purposes for which modern computers can be put to use.

Reference:

1. Computational Thinking for the Modern Problem Solver, By Riley DD, Hunt K.A CRC press, 2014
2. Ferragina P, Luccio F. Computational Thinking: First Algorithms, Then Code. Springer

Web References:

http://www.flowgorithm.org/documentation/

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Activity – 1 from Part A	Write up on the activity/ task	5
	Demonstration of the activity/ task	10
Activity-2 from Part B	Write up on the activity/ task	5
	Demonstration of the activity/ task	10
Viva Voice based on Lab Activities		05
Practical Records		05
Total		**40**

Course Code: CAC02	Course Title: Programming in C
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Course Content

Content	Hours
Unit – 1	
Introduction to C Programming: Overview of C; History and Features of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. **C Programming Basic Concepts:** C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. **Input and output with C:** Formatted I/O functions - *printf* and *scanf,* control stings and escape sequences, output specifications with *printf* functions; Unformatted I/O functions to read and display single character and a string - *getchar, putchar, gets* and *puts* functions.	10
Unit – 2	
C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion. **Control Structures:** Decision making Statements - S*imple if, if_else, nested if_else, else_if ladder, Switch Case*, *goto, break* & *continue* statements; Looping	12

Statements - Entry controlled and exit controlled statements, *while, do-while, for* loops, Nested loops.	
Unit – 3	
Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. **Strings:** Declaring & Initializing string variables; String handling functions - *strlen, strcmp, strcpy and strcat;* Character handling functions - *toascii, toupper, tolower, isalpha, isnumeric* etc.	8
Unit – 4	
Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;	4
Unit – 5	
User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. **User defined data types:** Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	8

Text Books:
1. C: The Complete Reference, By Herbert Schildt.
2. C Programming Language, By Brain W. Kernighan
3. Kernighan & Ritchie: The C Programming Language (PHI)

Reference Books:
1. P. K. Sinha & Priti Sinha: Computer Fundamentals (BPB)
2. E. Balaguruswamy: Programming in ANSI C (TMH)
3. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
4. V. Rajaraman: Programming in C (PHI – EEE)
5. S. Byron Gottfried: Programming with C (TMH)
6. Yashwant Kanitkar: Let us C
7. P.B. Kottur: Programming in C (Sapna Book House)

Course Code: CAC02P	**Course Title: C Programming Lab**
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks:10
Exam Marks: 40	Exam Duration: 04

Programming Lab

Part A:

1. Program to read radius of a circle and to find area and circumference
2. Program to read three numbers and find the biggest of three
3. Program to demonstrate library functions in math.h
4. Program to check for prime
5. Program to generate n primes
6. Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8. Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9. Program to find the roots of quadratic equation (demonstration of switch Case statement)
10. Program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11. Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

Part B:

1. Program to find the length of a string without using built in function
2. Program to demonstrate string functions.
3. Program to demonstrate pointers in C
4. Program to check a number for prime by defining isprime( ) function
5. Program to read, display and to find the trace of a square matrix
6. Program to read, display and add two m x n matrices using functions
7. Program to read, display and multiply two m x n matrices using functions

8. Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.

9. Program to Reverse a String using Pointer

10. Program to Swap Two Numbers using Pointers

11. Program to demonstrate student structure to read & display records of n students.

12. Program to demonstrate the difference between structure & union.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		**40**

Course Code: CAC03(a)	**Course Title:** Mathematical Foundation
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03

Course Outcomes (COs):

- Study and solve problems related to connectives, predicates and quantifiers under different situations.
- Develop basic knowledge of matrices and to solve equations using Cramer's rule.
- Know the concept of Eigen values.
- To develop the knowledge about derivatives and know various applications of differentiation.
- Understand the basic concepts of Mathematical reasoning, set and functions

Content	Hours
Unit – 1	
Basic concepts of set theory: Mathematical logic introduction-statements Connectives-negation, conjunction, disjunction- statement formulas and truth tables- conditional and bi Conditional statements- tautology contradiction-equivalence of formulas-duality law-Predicates and Quantifiers, Arguments.	10
Unit – 2	
Operations on sets: power set- Venn diagram Cartesian product-relations - functions- types of functions - composition of functions.	10
Unit – 3	
Matrix algebra: Introduction-Types of matrices-matrix operations-transpose of a matrix -determinant of matrix - inverse of a matrix- Cramer's rule	10
Unit – 4	
Matrix: finding rank of a matrix - normal form-echelon form cayley Hamilton theorem-Eigen values	6
Unit – 5	
Differential calculus: Functions and limits - Simple Differentiation of Algebraic Functions – Evaluation of First and Second Order Derivatives – Maxima and Minima	6

Text Books:
P. R. Vittal-Business Mathematics and Statistics, Margham Publications, Chennai,
Reference Books:
B. S. Vatsa-Discrete Mathematics –New Age International Limited Publishers, New Delhi

Course Code: CAC03(b)	Course Title: Accountancy
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03

Course Outcomes (COs):

- Study and understand Accounting, systems of Book, Branches of accounting advantage and limitations
- Know the concept of accounting, financial accounting process and Journalization
- Maintenance different account book and reconciliations
- Preparations of different bills, and trial balance.
- Understand the basic concepts of Mathematical reasoning, set and functions

Content	Hours
Unit – 1	
Introduction: History and Development of Accounting, Meaning, Objectives and functions of Accounting, Book keeping V/s Accounting, Users of accounting data, systems of book keeping and accounting, branches of accounting, advantages and limitations of accounting	10
Unit – 2	
Accounting Concepts and Convention: Meaning, need and classification, accounting standards meaning, need and classification of Indian accounting standards. Accounting principles V/s accounting standard Financial Accounting Process: Classification of accounting transactions and accounts, rules of debit and credit as per Double Entry System. Journalization and Ledger posting.	10
Unit – 3	
Preparation of Different Subsidiary Books: Purchase Day book Sales Day Book, Purchase Returns Day Book, Sales Returns Day Book, Cash Book. Bank Reconciliation Statement: Meaning, Causes of Difference, Advantages, Preparation of Bank Reconciliation Statements.	10
Unit – 4	
Account Procedure: Honor of the Bill, Dishonor of the Dill, Endorsement, Discounting, Renewal, Bill for collection, Retirement of the Bill, Accommodation	6

Bills, Bill Receivable Book and Payable Book. Preparation of Trial Balance: Rectification of errors and Journal Proper	
Unit – 5	
Preparation of Final Accounts: Meaning, need and classification, Preparation of Manufacturing, Trading, Profit and loss account and Balance – Sheet of sale-traders and partnership firms.	6

Text Books:

1. S. Ramesh, B.S. Chandrashekar, A Text Book of Accountancy.
2. V.A. Patil and J.S. Korihalli, Book – keeping and accounting, (R. Chand and Co. Delhi).
3. R. S. Singhal, Principles of Accountancy, (Nageen Prakash pvt. Lit. Meerut).
4. M. B. Kadkol, Book – Keeping and Accountancy, (Renuka Prakashan, Hubil)
5. Vithal, Sharma:Accounting for Management, Macmillan Publishers, Mumbai.

Reference Books:

1. B.S. Raman, Accountancy, (United Publishers, Mangalore).
2. Tulsian, Accouning and Finacial Management – I: Financial Accounting – Person Education.

Course Code: CAC04	**Course Title: Data Structures using C**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
- Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing

Course Content

Content	Hours
Unit – 1	
Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - *malloc*, *calloc*, *realloc* and *free.* Algorithm Specification, Performance Analysis, Performance Measurement Recursion: Definition; Types of recursions; Recursion Technique Examples - GCD, Binomial coefficient ^nC_r, Towers of Hanoi; Comparison between iterative and recursive functions.	8
Unit – 2	
Arrays: Basic Concepts – Definition, Declaration, Initialisation, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching - Sequential Search,	10

Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.	
Unit – 3	
Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Header liked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection,	8
Unit – 4	
Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls. **Queues**: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;	8
Unit – 5	
Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; **Binary tree:** Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal;	8

Text Books
   1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures

References

   1. Tanenbaum: Data structures using C (Pearson Education)
   2. Kamathane: Introduction to Data structures (Pearson Education)
   3. Y. Kanitkar: Data Structures Using C (BPB)
   4. Kottur: Data Structure Using C
   5. Padma Reddy: Data Structure Using C
   6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007))

Course Code: CAC04P	Course Title: Data Structures Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 03 Hours

Programming Lab

Part A:

1. Program to find GCD using recursive function
2. Program to display Pascal Triangle using binomial function
3. Program to generate n Fibonacci numbers using recursive function.
4. Program to implement Towers of Hanoi.
5. Program to implement dynamic array, find smallest and largest element of the array.
6. Program to create two files to store even and odd numbers.
7. Program to create a file to store student records.
8. Program to read the names of cities and arrange them alphabetically.
9. Program to sort the given list using selection sort technique.
10. Program to sort the given list using bubble sort technique.

Part B:

1. Program to sort the given list using insertion sort technique.
2. Program to sort the given list using quick sort technique.
3. Program to sort the given list using merge sort technique.
4. Program to search an element using linear search technique.
5. Program to search an element using recursive binary search technique.
6. Program to implement Stack.
7. Program to convert an infix expression to postfix.
8. Program to implement simple queue.
9. Program to implement linear linked list.
10. Program to display traversal of a tree.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		**40**

Course Code: CAC05	**Course Title: Object Oriented Programming with JAVA**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand the features of Java and the architecture of JVM
- Write, compile, and execute Java programs that may include basic data types and control flow constructs and how type casting is done
- Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concepts of polymorphism andinheritance
- The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA's Exceptional handling mechanism compared to other Programming Language
- Write, compile, execute Java programs that include GUIs and event driven programming and also programs based on files

Course Content

Content	Hours
Unit – 1	
Introduction to Java: Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.	6
Unit – 2	
Objects and Classes: Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference.	6
Unit – 3	
Inheritance and Polymorphism: Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	8
Unit – 4	
Event and GUI programming: Event handling in java, Event types, Mouse and	10

key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing, Exceptional handling mechanism.	

Unit – 5

I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	6

Unit – 6

Multithreading in java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally, Collections in java, Introduction to JavaBeans and Network Programming.	6

Text Books

1. Programming with Java, By E Balagurusamy – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.

2. Core Java Volume I – Fundamentals, By Cay S. Horstmann, Prentice Hall

3. Object Oriented Programming with Java : Somashekara, M.T., Guru, D.S., Manjunatha, K.S

Reference Books:

1. Java 2 - The Complete Reference – McGraw Hill publication.

2. Java - The Complete Reference, 7th Edition, By Herbert Schildt– McGraw Hill publication.

Course Code: CAC05P	**Course Title:** JAVA Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 04 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Implement Object Oriented programming concept using basic syntaxes of control Structures
- Identify classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem
- Demonstrates how to achieve reusability using inheritance
- Demonstrate understanding and use of interfaces, packages, different exception handling mechanisms and concept of multithreading for robust faster and efficient application development.
- Identify and describe common user interface components to design GUI in Java using Applet & AWT along with response to events

Practice Lab

1. Program to print the following triangle of numbers
   1
   1 2
   1 2 3
   1 2 3 4
   1 2 3 4 5
2. Program to simple java application, to print the message, "Welcome to java"
3. Program to display the month of a year. Months of the year should be held in an array.
4. Program to find the area of rectangle.
5. program to demonstrate a division by zero exception
6. Program to create a user defined exception say Pay Out of Bounds.

Programming Lab

PART A: Java Fundamentals OOPs in Java

1. Program to assign two integer values to X and Y. Using the 'if' statement the output of the program should display a message whether X is greater than Y.
2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)
3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.

4. Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.

5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.

6. Program
   a. To find the area and circumference of the circle by accepting the radius from the user.
   b. To accept a number and find whether the number is Prime or not

7. Program to create a student class with following attributes;
   Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, TotalMarks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.

8. In a college first year class are having the following attributes Name of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class

9. Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class

10. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.

11. Create a package 'student. Fulltime. BCA 'in your current working directory
    a. Create a default class student in the above package with the following attributes: Name, age, sex.
    b. Have methods for storing as well as displaying

PART B: Exception Handling & GUI Programming

1. Program to catch Negative Array Size Exception. This exception is caused when thearray is initialized to negative values.
2. Program to handle Null Pointer Exception and use the "finally" method to display a message to the user.
3. Program which create and displays a message on the window
4. Program to draw several shapes in the created window

5. Program to create an applet and draw grid lines
6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
7. Create a frame which displays your personal details with respect to a button click
8. Create a simple applet which reveals the personal information of yours.
9. Program to move different shapes according to the arrow key pressed.
10. Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night
11. Demonstrate the various mouse handling events using suitable example.
12. Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		**40**

Course Code: CAC06	**Course Title: Discrete Mathematical Structures**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- To understand the basic concepts of Mathematical reasoning, set and functions.
- To understand various counting techniques and principle of inclusion and exclusions.
- Understand the concepts of various types of relations, partial ordering and
- equivalence relations.
- Apply the concepts of generating functions to solve the recurrence relations.
- Familiarize the fundamental concepts of graph theory and shortest path algorithm

Course Content

Content	Hours
Unit – 1	
The Foundations: Logic and proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy. **Basic Structures**: Sets, Functions, Sequences, Sums, and Matrices: Sets, set operations, Functions, Sequences and Summations, matrices.	12
Unit – 2	
Counting: Basics of counting, Pigeonhole principle, Permutation and combination, Binomial Coefficient and Combination, Generating Permutationand Combination**.** **Advanced Counting Techniques**: Applications of Recurrence Relations, Solving Linear Recurrence, Relations, Divide and Conquer Algorithms and Recurrence Relations, Generating functions, Inclusion-Exclusion, Applications ofInclusion-exclusion.	10
Unit – 3	
Induction and Recursion: Mathematical Induction, Strong Induction and Well-	12

Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Corrections. **Relation:** Properties of relation, Composition of relation, Closer operation on relation, Equivalence relation and partition. Operation on relation, Representing relation.	
Unit – 4	
Graphs: Graphs and Graph models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.	8

Text Book:

1. Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition, 2012.

References:

2. Discrete Mathematical Structure, Bernard Kolman, Robert C, Busby, Sharon Ross, 2003.

3. Graph Theory with Applications to Engg and Comp. Sci: Narsingh Deo-PHI 1986.

4. Discrete and Combinatorial Mathematics Ralph P. Grimaldi, B. V. Ramatta, Pearson, Education, 5 Edition.

5. Discrete Mathematical Structures, Trembley and Manobar.

Computer Application Core Courses (CA C) for BCA (Hons)

Sl. No	Course Code	Title of the Paper
1	CAC01	Fundamentals of Computers
2	CAC02	Programming in C
3	CAC03 (a)/(b)	Mathematical Foundation/ Accountancy
4	CAC04	Discrete Mathematical Structures
5	CAC05	Object Oriented Concepts using JAVA
6	CAC06	Data Structures using C
7	CAC07	Data Base Management Systems
8	CAC08	C# and DOT NET Framework
9	CAC09	Computer Communication and Networks
10	CAC10	Python Programming
11	CAC11	Computer Multimedia and Animation
12	CAC12	Operating System Concepts
13	CAC13	Internet Technologies
14	CAC14	Statistical Computing and R Programming
15	CAC15	Software Engineering
16	CAC16	PHP and MySQL
17	CAC17	Artificial Intelligence and Applications
18	CAC18	Analysis and Design of Algorithms
19	CAC19	Data Mining and Knowledge Management
20	CAC20	Automata Theory and Compiler Design
21	CAC21	Cryptography and Network Security

Computer Application Electives (CA E) for BCA (Hons)

Sl. No	Computer Application Electives (CA E)
1	Business Intelligence
2	Cyber Law and Cyber Security
3	Data Analytics
4	Data Compression
5	Embedded Systems
6	Fundamentals of Data Science
7	Internet of Things (IoT)
8	Machine Learning
9	Mobile Application Development
10	Open-source Programming
11	Pattern Recognition
12	Storage Area Networks

Vocational Electives

Sl. No	Vocational Electives
1	DTP, CAD and Multimedia
2	Hardware and Server Maintenance
3	Web Content Management Systems
4	Computer Networking
5	Health Care Technologies
6	Digital Marketing
7	Office Automation

Open Electives in Computer Science:

(For Students studying Core Courses other than Computer Science/ Computer Applications)

Sl. No	Open Electives in Computer Science
1	C Programming Concepts
2	Office Automation
3	Multimedia Processing
4	Python Programming Concepts
5	R Programming
6	E-Content Development
7	E-Commerce
8	Web Designing
9	Computer Animation
10	Accounting Package

Curriculum for III BCA & IV BCA

Sem	Core Courses	Hour / Week		DS Elective Courses	Hours/ Week
		Theory	Lab		
III	Database Management Systems	3			
	C# and DOT NET Framework	3			
	Computer Communication andNetworks	3			
	LAB: DBMS		4		
	LAB: C# and DOT NET Framework		4		
IV	Python Programming	3			
	Computer Multimedia andAnimation	3			
	Operating Systems Concepts	3			
	LAB: Multimedia and Animation		4		
	LAB: Python programming		4		

Course Title: **Database Management System**	Course code: 21BCA3C7L
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the various database concepts and the need for database systems.
- Identify and define database objects, enforce integrity constraints on a database usingDBMS.
- Demonstrate a Data model and Schemas in RDBMS.
- Identify entities and relationships and draw ER diagram for a given real-worldproblem.
- Convert an ER diagram to a database schema and deduce it to the desired normalform.
- Formulate queries in Relational Algebra, Structured Query Language (SQL) fordatabase manipulation.
- Explain the transaction processing and concurrency control techniques.

DSC7: Database Management System (DBMS)

Unit	Description	Hours
1	**Database Architecture:** Introduction to Database system applications. Characteristics and Purpose of database approach. People associated with Database system. Data models. Database schema. Database architecture. Data independence. Database languages, interfaces, and classification of DBMS.	08
2	**E-R Model:** Entity-Relationship modeling: E – R Model Concepts: Entity, Entity types, Entity sets, Attributes, Types of attributes, key attribute, and domain of an attribute. Relationships between the entities. Relationship types, roles and structural constraints, degree and cardinality ratio of a relationship. Weak entity types, E -R diagram.	08

3	Relational Algebra: Basic Relational Algebra operations. Set theoretical operations on relations. JOIN operations Aggregate Functions and Grouping. Nested Sub Queries-Views. Introduction to PL/SQL & programming of above operations in PL/SQL	10
4	**Data Normalization:** Anomalies in relational database design. Decomposition. Functional dependencies. Normalization. First normal form, Second normal form, Third normal form. Boyce-Codd normal form.	07
5	**Query Processing Transaction Management:** Introduction Transaction Processing. Single user & multiuser systems. Transactions: read & write operations. Need of concurrency control: The lost update problem, Dirty read problem. Types of failures. Transaction states. Desirable properties (ACID properties) of Transactions. Concurrency Control Techniques: Locks and Time stamp Ordering. Deadlock & Starvation.	09

References:
1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7th Edition,Pearson, 2015
2. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
3. Introduction to Database System, C J Date, Pearson, 1999.
4. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6thEdition, McGraw Hill, 2010.
5. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition,McGraw Hill, 2002

Practical's:

Student would be able to create tables, execute queries and PL/SQL programs.

1. Execute a single line query and group functions.
2. Execute DDL Commands.
3. Execute DML Commands
4. Execute DCL and TCL Commands.
5. Implement the Nested Queries.
6. Implement Join operations in SQL
7. Create views for a particular table
8. Implement Locks for a particular table
9. Write PL/SQL procedure for an application using exception handling.
10. Write PL/SQL procedure for an application using cursors.
11. Write a PL/SQL procedure for an application using functions

Write a PL/SQL procedure for an application using package

Course Title: **C# and Dot Net Framework**	Course code: 21BCA3C8L
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:
- Describe Object Oriented Programming concepts like Inheritance and Polymorphismin C# programming language.
- Interpret and Develop Interfaces for real-time applications.
- Build custom collections and generics in C#.

DSC8: C# and Dot Net Framework

Unit	Description	Hours
1	**Introduction to .Net Technologies:** Introduction to Web Technologies, HTML Basics, Scripts. Sample Programs. Advantages and Disadvantages of Client-side and Server-side Scripts. Overview of Client-side Technologies and Server-side Technologies.	08
2	**Introduction to C#:**Overview of C#, Literals, Variables, Data Types, Operators, Expressions, Control Structures-Methods, Arrays, Strings, Structures, Enumerations. **OOPS with C#:** Classes, Objects, Inheritance, Polymorphism, Interfaces, Operator Overloading Delegates, Events, Errors and Exceptions.	08
3	**Introduction to VB.NET**: Introduction VB.NET -IDE – Creating a shortcut to start VB.NET. Maneuverings the Toolbar Auto-hide, Docking and Undocking, Placing and Resizing the Windows, Forms, Properties Window and Solution Explorer. Writing and Event Procedure. Execution Basic Keywords. Data Types. VB.NET statements. Conditional statements: If Else, Select Case, Switch and Choose Loops: Do, For Next, For Each Next, While loop. Arrays.	08
4	**Application Development on .NET:**C#.NET: Building Windows Applications, VB.NET: Windows Forms. Working with Controls, Timer, Picture-box, Group-box, Combo-box, Horizontal and Vertical Scrollbar, Numeric-up-down, Track-bar, and Progress-bar. Subroutines and Functions inVB.NET. Database applications	10

5	**ADO .NET Connectivity:** Introduction to ADO.NET, ADO vs ADO.NET. Architecture: Data reader, Data adopter, Accessing Data with ADO.NET Programming Web Applications with Web Forms. ASP .NET applications with ADO.NET	08

References:
1. "Programming in C#", E. Balagurusamy, 4th Edition, Tata McGraw-Hill, 2017.
2. "Visual Basic.NET", Shirish Chavan, 3rd Edition, Pearson Education, 2009.
3. "ASP.NET and VB.NET Web Programming", Matt J. Crouch, Edition 2012.
4. "Computing with C# and the .NET Framework", Arthur Gittleman, 2nd Edition, Jones &Bartlett Publishers, 2011

Practicals:

1. Develop a C# .NET console application to demonstrate the conditional statements.
2. Develop a C# .NET console application to demonstrate the control statements.
3. Develop an application in C#.NET that demonstrates the windows controls
4. Demonstrate Multithreaded Programming in C#.NET
5. Demonstrate subroutines and functions in C#.NET
6. Develop an application for deploying various built-in functions in VB.NET
7. Develop an MDI application for Employee Pay-roll transactions in VB.NET
8. Construct a console application to demonstrate the OOP Concepts
9. Develop a web application in VB.NET for dynamic Login Processing
10. Develop a Windows application with database connectivity for core-bankingtransactions

Course Title: **Computer Communicationand Networks**	Course code: 21BCA3C9L
Total Contact Hours: 42	Course Credits: 03
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:
- Explain the transmission technique of digital data between two or more computersand a computer network that allows computers to exchange data.
- Apply the basics of data communication and various types of computer networks inreal world applications.
- Compare the different layers of protocols.
- Compare the key networking protocols and their hierarchical relationship in theconceptual model like TCP/IP and OSI.

DSC9: Computer Communication and Networks

Unit	Description	Hours
1	**Introduction:** Computer Networks and its applications, Network structure, network architecture, Topologies, LAN, WAN, MAN, The OSI referencemodel, The TCP/IP reference model.	08
2	**The Physical Layer:** Transmission Media – Twisted pair, coaxial cable, optical fiber, radio transmission, microwaves and infrared transmission, Switching – message switching, Multiplexing.	07
3	**The Data Link Layer:** Data Link Layer design issues, Error detection – Single parity checking, Checksum, polynomial codes – CRC, Error correction- Hamming code, Elementary data link protocols, sliding window protocols	08
4	**The Network Layer:** Network layer design issues, Routing algorithms – Flooding, Distance vector routing, Hierarchical routing, Link state routing, Congestion, control algorithms – Leaky bucket, token bucket algorithm, admission control, Hop by Hop choke packets.	09
5	**The Transport Layer and Application Layer:** Elements of Transport service, Elements of Transport, protocols, Internet transport protocols (TCP &UDP), DNS, Electronic Mailing, and World Wide Web.	10

References:

1. Computer Networks, Andrew S. Tanenbaum, 5th Edition, Pearson Education, 2010. Data Communication & Networking, Behrouza A Forouzan, 3rd Edition, Tata McGraw Hill,2001.
2. 3. Data and Computer Communications, William
3. Data Communication and Computer Networks, Brijendra Singh, 3rd Edition, PHI, 2012.
4. Data Communication & Network, Dr. Prasad, Wiley Dreamtech.
5. http://highered.mheducation.com/sites/0072967757/index.htmls

Course Title: **Python Programming**	Course code: 21BCA3C10L
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the basic concepts of Python Programming.
- Demonstrate proficiency in the handling of loops and creation of functions.
- Identify the methods to create and manipulate lists, tuples and dictionaries.
- Discover the commonly used operations involving file handling.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Develop the emerging applications of relevant fields using Python.

DSC10: Python Programming

Unit	Description	Hours
1	**Introduction** to Features and Applications of Python; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program. **Python Basics:** Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built-in Functions- Console Input and Console Output, Type Conversions; Python Libraries; Importing Libraries with Examples. Python Control Flow: Types of Control Flow; Control Flow Statements- if, else, elif, while loop, break, continue statements, for loop Statement; range () and exit () functions.	08
2	**Exception Handling:** Types of Errors; Exceptions; Exception Handling usingtry, except and finally. **Python Functions:** Types of Functions; Function Definition- Syntax Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Recursive Functions; Scope and Lifetime of Variables in Functions. **Strings:** Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifiers; Escape Sequences; Raw and Unicode Strings; Python String Methods.	08

3	**Lists:** Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists. Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries. **Tuples and Sets:** Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Built-inFunctions on Sets; Set Methods.	08
4	**File Handling:** File Types; Operations on Files– Create, Open, Read, Write, Close Files; File Names and Paths; Format Operator. **Object Oriented Programming:** Classes and Objects; Creating Classes and Objects; Constructor Method; Classes with Multiple Objects; Objects as Arguments; Objects as Return Values; Inheritance- Single and Multiple Inheritance, Multilevel and Multipath Inheritance; Encapsulation-Definition, Private Instance Variables; Polymorphism- Definition, Operator Overloading.	08
5	**GU Interface:** The tkinter Module; Window and Widgets; Layout Management- pack, grid and place. **Python SQLite:** The SQLite3 module; SQLite Methods- connect, cursor, execute, close; Connect to Database; Create Table; Operations on Tables-Insert, Select, Update. Delete and Drop Records. **Data Analysis:** NumPy- Introduction to NumPy, Array Creation using NumPy, Operations on Arrays; Pandas- Introduction to Pandas, Series and DataFrames, Creating DataFrames from Excel Sheet and .csv file, Dictionary and Tuples. Operations on DataFrames. **Data Visualisation:** Introduction to Data Visualisation; Matplotlib Library; Different Types of Charts using Pyplot- Line chart, Bar chart and Histogram and Pie chart.	10

References:
1. Think Python How to Think Like a Computer Scientist, Allen Downey et al., 2n Edition, Green Tea Press. Freely available online @ https://www.greenteapress.com/thinkpython/thinkCSpy.pdf, 2015.
2. Introduction to Python Programming, Gowrishankar S et al., CRC Press, 2019.
3. Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and thePython Programming Language, Fabio Nelli, Apress®, 2015
4. Advance Core Python Programming, MeenuKohli, BPB Publications, 2021.
5. Core PYTHON Applications Programming, Wesley J. Chun, 3rd Edition, Prentice Hall2012.
6. Automate the Boring Stuff, Al Sweigart, No Starch Press, Inc, 2015.
7. Data Structures and Program Design Using Python, D Malhotra et al., MercuryLearning and Information LLC, 2021.
8. http://www.ibiblio.org/g2swap/byteofpython/read/
9. https://docs.python.org/3/tutorial/index.html

Programs for Practical Component:

Part-A
1. Check if a number belongs to the Fibonacci Sequence
2. Solve Quadratic Equations
3. Find the sum of n natural numbers
4. Display Multiplication Tables
5. Check if a given number is a Prime Number or not
6. Implement a sequential search
7. Create a calculator program
8. Explore string functions
9. Implement Selection Sort
10. Implement Stack
11. Read and write into a file

Part-B
1. Demonstrate usage of basic regular expression
2. Demonstrate use of advanced regular expressions for data validation.
3. Demonstrate use of List
4. Demonstrate use of Dictionaries
5. Create SQLite Database and Perform Operations on Tables
6. Create a GUI using Tkinter module
7. Demonstrate Exceptions in Python
8. Drawing Line chart and Bar chart using Matplotlib
9. Drawing Histogram and Pie chart using Matplotlib
10. Create Array using NumPy and Perform Operations on Array
11. Create DataFramefrom Excel sheet using Pandas and Perform Operations onDataFrames

Note: A minimum of 10 Programs should be done in each Part.

Evaluation Scheme for Lab Examination:

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	03
	Execution and Formatting	07
Program -2 from Part B	Writing the Program	03
	Execution and Formatting	07
Viva Voice based on Python Programming		05
Total		25

Course Title: **Computer Multimedia &Animation**	Course code: 21BCA3C11L
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course the student should be able to:

- Understand basic elements using in web development
- Understand and develop animations using CSS
- Understand and develop HTML5-SVG animation

- Understand and develop HTML5-CANVAS animations

DSC11: Computer Multimedia & Animation

Unit	Description	Hours
1	Web Design: Origins and evolution of HTML, Basic syntax, Basic text markup, Images, Lists, Tables, Forms, Frame, Overview and features of HTML5. CSS: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The <span> and <div> tags; Overview and features of CSS3. JavaScript: Object orientation and JavaScript; General syntactic characteristics; Primitives, operations, and expressions; Screen output and keyboard input.	10
2	Animation: What is an Animation? The Start and End States, Interpolation, Animations in HTML. All About CSS Animations, Creating a Simple Animation, Detailed Look at the CSS Animation Property, Keyframes, Declaring Multiple Animations, Wrap-up. All About CSS Transitions, Adding a Transition, Looking at Transitions in Detail, The Longhand Properties, Longhand Properties vs. Shorthand Properties, Working with Multiple Transitions.	09
3	HTML5 – SVG: Viewing SVG Files, Embedding SVG in HTML5, HTML5 – SVG Circle, HTML5 – SVG Rectangle, HTML5 – SVG Line, HTML5 – SVG Ellipse, HTML5 – SVG Polygon, HTML5 – SVG Polyline, HTML5 – SVG Gradients, HTML5 – SVG Star.	08
4	HTML5 – CANVAS: The Rendering Context, Browser Support, HTML5 Canvas Examples, Canvas - Drawing Rectangles, Canvas - Drawing Paths, Canvas - Drawing Lines, Canvas - Drawing Bezier Curves, Canvas - DrawingQuadratic Curves, Canvas - Using Images, Canvas - Create Gradients,	08

5	HTML5 - Styles and Colors, Canvas - Text and Fonts, Canvas - Pattern andShadow, Canvas - Save and Restore States, Canvas - Translation, Canvas - Rotation, Canvas - Scaling, Canvas - Transforms, HTML5 Canvas -Composition, Canvas – Animations.	07

References:

1. The Complete Reference HTML and CSS, 5th Edition, Thomas A Powell, 2017.
2. Animation in HTML, CSS, and JavaScript, Kirupa Chinnathambi,Createspace Independent Pub, 2013.
3. https://www.w3.org/Style/CSS/current-work#CSS3
4. http://bedford-computing.co.uk/learning/cascading-style-sheets-css/

Practicals

<u>Part A:</u>
1) Write a HTML/5 program to demonstrate the use of Font family, font variant, font style, and font size.
2) Write a HTML/5 program to display random contents using list properties:
   a) Ordered list b) Unordered list
3) Write a HTML/5 program to create gradient using CSS.
4) Write a HTML/5 code to demonstrate following CSS animation properties:
   a) Delay   b) Direction   c) Duration
5) Write a HTML/5 program to demonstrate key frames
6) Write a HTML/5 code to demonstrate CSS transition and transformation.
7) Write a HTML/5 program to turn on/off a light bulb using JavaScript. Make use of .gif image and buttons.

<u>Part B:</u>
1) Write a HTML/5 program to draw rectangle, line, polygon, polyline using SVG.
2) Write a HTML/5 program to draw linear and radial gradient ellipse using SVG.
3) Write a HTML/5 program to draw a star using SVG.
4) Write a HTML/5 program to draw line, circle, rectangle, gradient, text using canvas.
5) Write a HTML/5 program to demonstrate translation, rotation, scaling, and transform using canvas.
6) Write a HTML/5 program to demonstrate Bezier Curves and Quadratic Curves.
7) Write a HTML/5 program to create canvas and add a red square onto the game area with up/down/left/right controller buttons.
8) Write a HTML/5 canvas program to add random size obstacles with red square controller button.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	3
	Execution and Formatting	7
Program – 2 from Part B	Writing the Program	3
	Execution and Formatting	7
Viva Voice based on Lab Activities		5
Total		**25**

Course Title: **Operating System Concepts**	Course code: 21BCA3C12L
Total Contact Hours: 42	Course Credits: 03
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the fundamentals of the operating system.
- Comprehend multithreaded programming, process management, processsynchronization, memory management and storage management.
- Compare the performance of Scheduling Algorithms
- Identify the features of I/O and File handling methods.

DSC12: Operating System Concepts

Unit	Description	Hours
1	**Introduction to Operating System:** Definition, History and Examples of Operating System; Computer System organization; Types of Operating Systems; Functions of Operating System; Systems Calls; Operating System Structure. **Process Management:** Process Concept- Process Definition, Process State, Process Control Block, Threads; Process scheduling- Multiprogramming, Scheduling Queues, CPU Scheduling, Context Switch; Operations onProcesses- Creation and Termination of Processes; Inter process communication (IPC)- Definition and Need for Inter process Communication; IPC Implementation Methods- Shared Memory and Message Passing;	08
2	**Multithreaded Programming**: Introduction to Threads; Types of Threads; Multithreading- Definition, Advantages; Multithreading Models; Thread Libraries; Threading Issues. **CPU Scheduling**: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling; Multiprocessor Scheduling; Real-Time CPU Scheduling.	10
3	**Process Synchronization:** Introduction; Race Condition; Critical Section Problem and Peterson's Solution; Synchronization Hardware; Semaphores; Classic Problems of Synchronization- Readers and Writers Problem, Dining Philosophers Problem; Monitors. **Deadlocks:** System Model; Deadlocks Characterization; Methods for Handling Deadlocks; Deadlock Prevention; Deadlock Avoidance; Deadlock Detection; and Recovery from Deadlock.	10

4	**Memory Management:** Logical and Physical Address Space; Swapping; Contiguous Allocation; Paging; Segmentation; Segmentation with Paging. **Virtual Memory:** Introduction to Virtual Memory; Demand Paging; Page Replacement; Page Replacement Algorithms; Allocation of frames; Thrashing.	08
5	**File System:** File Concepts- Attributes, Operations and Types of Files; File System; File Access methods; Directory Structure; Protection; File System Implementation- File System Structure, Allocation Methods, Free Space Management	06

References:
1. Operating System Concepts, Silberschatz' et al., 10thEdition, Wiley, 2018.
2. Operating System Concepts - Engineering Handbook, Ghosh PK, 2019.
3. Understanding Operating Systems, McHoes A et al., 7th Edition, Cengage Learning,2014.
4. Operating Systems - Internals and Design Principles, William Stallings, 9th Edition,Pearson.
5. Operating Systems – A Concept Based Approach, Dhamdhere, 3rd Edition, McGrawHill Education India.
6. Modern Operating Systems, Andrew S Tanenbaum, 4th Edition, Pearson.

Course Title: **Open Source Tools**	Course Credits: 2 (1L+0T+2P)
Semester: III	Duration of SEE: 01 Hour
Total Contact Hours: 13 hours of theory and 26-28hours of practical's	SEE: 30 Marks IA: 20 Marks

Course Outcomes (COs):
- Recognize the benefits and features of Open Source Technology and to interpret,contrast and compare open source products among themselves
- Use appropriate open source tools based on the nature of the problem
- Write code and compile different open-source software.

Course Content (Open Source Tools)

Module	Details of topic	Duration
Module 1: Open Source Softwares	i. Introduction to Open sources, Need of Open Sources, Open Source –Principles, Standard Requirements, Advantages of Open Sources – ii. Free Software – FOSS iii. Licenses – GPL, LGPL, Copyrights, Patents, Contracts & Licenses and Related Issues iv. Application of Open Sources. Open SourceOperating Systems : FEDORA, UBUNTU	05 hours
Module 2: Programmin gTools And Techniques	i. Usage of design Tools like Argo UML or equivalent ii. Version Control Systems like Git or equivalent iii. Bug Tracking Systems (Trac, BugZilla) iv. BootStrap	04 hours
Module 3: Case Studies	**i.** Apache **ii.** Berkeley Software Distribution **iii.** Mozilla (Firefox) **iv.** Wikipedia **v.** Joomla **vi.** GNU Compiler Collection **vii.** Libre Office	04 hours

Text Book:
  1. KailashVadera, Bhavyesh Gandhi, "Open Source Technology", Laxmi Publications Pvt.Ltd 2012, 1st Edition.

Reference Book:
  1. Fadi P. Deek and James A. M. McHugh, "Open Source: Technology and Policy",Cambridge Universities Press 2007.

Course Title: **Python Programming Concepts**	Course Credits: 3 (3L+0T+0P)
Semester: III	Duration of SEE: 03 Hour
Total Contact Hours: 42	SEE: 60 Marks IA: 40 Marks

Course Outcomes (COs):

- Explain the fundamentals of Computers.
- Explain the basic concepts of Python Programming.
- Demonstrate proficiency in the handling of loops and the creation of functions.
- Identify the methods to create and store strings.

Unit I   Fundamentals of Computers   10 Hrs

Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organization of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs
– Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.

Unit II   Python Basics   10 Hrs

Introduction to Features and Applications of Python; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program. Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built-in Functions- Console Input and Console Output, Type Conversions; Python Libraries; Importing Libraries with Examples; Illustrative programs.

Unit III   08 Hrs

Python Control Flow: Types of Control Flow; Control Flow Statements- if, else, elif, while loop, break, continue statements, for loop Statement; range() and exit () functions; Illustrativeprograms.

Unit IV   08 Hrs

Python Functions: Types of Functions; Function Definition- Syntax, Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Illustrative programs.

Unit V   06 Hrs

Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifiers; Escape Sequences; Raw and Unicode Strings; Python String Methods; Illustrative programs.

References

1. Computer Fundamentals (BPB), P. K. Sinha & Priti Sinha
2. Think Python How to Think Like a Computer Scientist, Allen Downey et al., 2ndEdition, Green Tea Press. Freely available online 2015. @https://www.greenteapress.com/thinkpython/thinkCSpy.pdf
3. Introduction to Python Programming, Gowrishankar S et al., CRC Press, 2019.
4. http://www.ibiblio.org/g2swap/byteofpython/read/
5. http://scipy-lectures.org/intro/language/python_language.html
6. https://docs.python.org/3/tutorial/index.html

Open Elective

PRINCIPLES OF INTERNET

Course Title: **PRINCIPLES OF INTERNET**	Course Credits: 3 (3L+0T+0P)
<mark>Semester: IV</mark>	Duration of SEE: 03 Hour
Total Contact Hours: 42	SEE: 60 Marks IA: 40 Marks

Course Outcomes (COs):
- Able to explain understand the internet standards and recent web technologies likeConferencing, newsgroup etc.
- Able to implement, compile, test and run HTML program
- Able to explain understand web Browsers.

UNIT I   **09 Hrs**

Basics of Networks – Advantages and disadvantages- Types of networks - LAN, WAN, MAN

-Topologies of Networks – Bus Topology, Ring Topology, Star Topology, Mesh Topology,

TreeTopology.

UNIT II   **06 Hrs**

Internet Basics – Advantages and disadvantages – Internet Applications

– IP Address- Internet Services– URL- E-mail - E-mail features – Emil Providers.

UNIT III   **09 Hrs**

Introduction to Internet Protocol –TCP –UDP-FTP – The Telnet –The Usenet – Internet Chat.

                                                                                                       09 Hrs
UNIT IV

WWW Overview – Web pages -The Web Servers – Web Browsers– Search

Engines-Introduction to Firewalls – Uses of Firewalls – Types of Firewalls.

UNIT V   **09 Hrs**

Art of creating a website – Hypertext and HTML – Features – Elements – Attributes- Structure

ofHTML– Basic tags in HTML – Formatting tags in HTML – HTML Lists –Types of Lists.

Text Book:

1. Raj Kamal , "Internet and Web Technologies", Mc GrawHill Education,2007.

References:
1. Joe krayank & Joe Habraken, "Internet 6 in 1", Prentice Hall of India Private Limited,New Delhi, 1998.
2. "Internet Complete", BPB publications, New Delhi, 1998.