DSEU

Delhi Skill and
Entrepreneurship University

Bachelor of Computer Application (BCA)

Syllabus Document

Effective from Academic Year 2021-22

Program Information

Program Vision

To nurture youth with global competence with knowledge in Computer Applications capable of driving their career in Computation and Information Technology.

Program Outcome:

What the students will learn by the end of program

1. Basic Knowledge/Skills: An Individual with a positive Attitude with the best Communication, Presentation, Leadership Skills, and Professional Competence.
2. Technical Knowledge/Skills: Problem-solving Capabilities with Logical Thinking
3. Software Skill and Project Skills: Program Development and Problem Solving Skills with Knowledge of Programming Language like C, C++, Python, Database Management, Computer Networking, etc
4. Personality Traits and Ethics
5. Soft Skills

Pedagogy and Teaching Methodology:

Three years of classroom training are interspersed with industry visits, guest lectures and paid apprenticeships. You will learn to gather order information, perform historical pattern analysis, follow up with trucking companies, communicate with clients, consolidate orders by destination, and monitor all consignments in real-time. Students will be exposed to the industry interface via internships, live projects, field placements, master classes by the industry personnel, and classroom online/ offline workshops and seminars. In the third year, students will be offered an apprenticeship in the industry under the guidance of industry experts and in-house faculty members. In the last semester, students will be able to consolidate the knowledge of all the five semesters in the apprenticeship, in order to find the best career path for themselves.

Credit scheme

Semester I

Sl No.	Course Code	Course Name	Hours/week				Total Credits
			L	T	O	P	
1	BCA-DC101	Mathematics - I (Calculus & Fourier Transformations)	3	1	0	0	4
2	BCA-DC102	Computer Organization and Architecture	2	1	0	2	4
3	BCA-DC103	Problem Solving & Programming in C	2	1	0	2	4
4	BCA-DC104	Information Technologies	2	1	0	2	4
5	BCA-DC105	Software Engineering	2	1	0	2	4
6	BCA-FW101	Face the World Skills-I	3	0	0	0	3
7	BCA-AE101	English Communication-I	2	0	0	0	2
Total			**16**	**5**	**0**	**8**	**25**

Semester II

Sl No.	Course Code	Course Name	Hours/week				Total Credits
			L	T	O	P	
1	BCA-DC201	Discrete Mathematics	3	1	0	0	4
2	BCA-DC202	Operating Systems	3	1	0	0	4
3	BCA-DC203	Free & Open Source Software (Linux Programming & Administration)	2	1	0	2	4
4	BCA-DC204	Data & File Structure using C	2	1	0	2	4
5	BCA-DC205	Object Oriented Programming using C++	2	1	0	2	4
6	BCA-AE201	Environmental Studies - I	2	0	0	0	2
7	BCA-AE202	English Communication-II	2	0	0	0	2
8	BCA-FW201	Face the World - II	3	0	0	0	3
Total			**19**	**5**	**0**	**6**	**27**

Semester III							
Sl No.	Course Code	Course Name	Hours/week				Total Credits
			L	T	O	P	
1	BCA-DC301	Web Technologies & E-Commerce	2	1	0	2	4
2	BCA-DC302	Computer Graphics	2	1	0	2	4
3	BCA-DC303	Python Programming	2	1	0	2	4
4	BCA-DC304	Database Management Systems	2	1	0	2	4
5	BCA-DC305	Computer Networks	2	1	0	2	4
6	FW-DG033	Face the World - III (Socio - Emotional Learning)	0	0	0	2	2
7	FW -DG043	Face the World - III (Design Thinking)	0	0	0	1	1
8	BCA-MC388	MOOC	0	0	2	0	1
9	AE-012	Environmental Studies-II	0	0	0	0	2
Total			**10**	**5**	**2**	**13**	**26**

BCA -MC388-1	Ethical Hacking
BCA -MC388-2	Programming, Data Structures And Algorithms Using Python

Semester IV							
Sl No.	Course Code	Course Name	Hours/week				Total Credits
			L	T	O	P	
1	BCA-DC401	Java Programming	2	1	0	2	4
2	BCA-DC402	Design and Analysis Algorithms	2	1	0	2	4
3	BCA-DC403	Introduction to Cloud Technologies	2	1	0	2	4
4	BCA-DC404	Cyber Security & Cyber Law	3	1	0	0	4
Students will be required to Select One of the Electives from Elective-I							
5	BCA-EC401	Elective - I Data Sciences Using Python	2	1	0	2	4
6	BCA-EC402	Elective - I C# Programming	2	1	0	2	4
7	BCA-EC403	Elective – I Administration of Cloud (Azure/Google Cloud/AWS)	2	1	0	2	4
8	BCA-EC404	Elective - I BlockChain Technologies	2	1	0	2	4
9	BCA-SM401	Seminar/ Research Paper Presentation	0	0	0	2	1
10	FW -DG064	Face the world skills (Entrepreneurship)	1	0	0	0	1
11	FW -DG034	Face the world skills (socio -Emotional Learning)	2	0	0	0	2
12	BCA-MC488	MOOC	0	0	2	0	1
Total			**14**	**5**	**2**	**10**	**25**

DSEU

SEMESTER I

BCA- DC101 | Mathematics- I (Calculus & Fourier Transformations)

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
3	1	0	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

1. To learn basic mathematical concepts.
2. To learn the transform for the use of Image Processing.
3. To learn the basics of transformation Techniques.

Syllabus:

Unit	Topics Covered	Hours
Unit 1	**Differential Calculus**	**10**
1.1	Functions, Elementary Functions, Domain, range, and algebra of Functions, Composite, even and odd Functions	
1.2	Limit, continuity and differentiability: Left and right-hand limit of a function, Algebra of Limits, Indeterminate Forms, L'Hospital Rule, Continuous Function, Properties of Continuous function, Continuity at a point, Continuity in an interval, Derivative of a function by the first principle, Differentiability of a function	
1.3	Differentiation: Differentiation of elementary functions algebraic, trigonometric, inverse trigonometric, logarithmic and exponential functions. Sum, product and quotient functions.	
1.4	Applications of Derivatives: Rate of change of quantities, applications of derivative in finding errors	
Unit 2	Integral Calculus	10
2.1	Basic Concept, definition, Properties of Integration, Integration as antiderivatives, Integration of Standard Functions, Methods of Integration By substitution, by parts, by partial functions, Definite Integral, some properties of definite integral	

2.2	Application of Integrals: Area under simple curve, Area of a region bounded by a curve and a line, Area between two curves	
Unit 3	Laplace Transform	10
3.1	Laplace transform of elementary functions, Shifting theorems, Transform of derivatives, Differentiation and Integration of transforms, Heaviside unit step and Dirac Delta functions	
3.2	Convolution theorem, Solution of ordinary linear differential equations used in Mechanics, Electric circuits and Bending of beams.	
Unit 4	Fourier Transform and Z Transform	10
4.1	Definition of Fourier transform, Fourier sine and cosine transforms, Fourier integral formula, Parseval's identity, Applications of Fourier transform in solving heat equations.	
4.2	Z transform: Definition, Linearity property, Z transform of elementary functions, Shifting theorems, Initial and final value theorem, Convolution theorem, Inversion of Z transforms, Solution of difference equations by Z transforms.	

References/suggested learning resources:

1. Kreyszig E., Advanced Engineering Mathematics, John Wiley, New York, 1999
2. Gerald, C.F., Wheatley P.O., Applied Numerical Analysis, Pearson, 2007.
3. Grewal B.S., Higher Engineering Mathematics, Khanna, New Delhi, 2000.
4. Jain R. K., Iyenger S.R.K., Advanced Engineering Mathematics, Narosa, 2002.
5. Jain R. K., Iyenger S.R.K., Jain M.K., Numerical Methods for Scientific and Engineering Computation, New Age International Publishers, 2012.

BCA- DC102 | Computer Organization and Architecture

Teaching Scheme			
Lecture hours per week	**Tutorial Hours per week**	**Practical hours per week**	**Credit**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

1. To learn the basic structure and operations of a computer.
2. To learn the arithmetic and logic unit and implementation of fixed-point and floating point arithmetic units.
3. To learn the basics of pipelined execution.
4. To understand parallelism and multi-core processors.
5. To understand the memory hierarchies, cache memories and virtual memories.
6. To learn the different ways of communication with I/O devices.

Syllabus:

Unit	Title	Hours
Unit 1	**Register Transfer Language and Computer Design**	**10**
1.1	Register Transfer and Micro-operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro Operations, Arithmetic logic shift unit	6
1.2	Basic Computer Organizations and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Register reference instructions, Input - Output Instructions, Design of Accumulator Logic	4
UNIT 2	**Design of CPU and Microprogrammed Control Unit**	**10**
2.1	Microprogrammed Control Unit: Control memory, address sequencing.	3
2.2	Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes. Difference between RISC and CISC.	3

2.3	Pipeline and Vector Processing: Arithmetic and Instruction pipeline, Vector operations, Matrix Multiplication, memory interleaving.	4
UNIT 3	**I/O Organization, Computer Arithmetic and Data transfer**	**10**
3.1	Input-Output Organization: Peripheral Devices, Input-Output Interfaces	**4**
3.2	Computer Arithmetic: Introduction, Multiplication Algorithms, Division Algorithms for fixed point-members.	**3**
3.3	Input-Output Organization: Peripheral devices, input-output interface, asynchronous data transfer, modes of data transfer, priority interrupt, direct memory access, input-output processor.	**3**
UNIT 4	**Memory Organization**	**10**
4.1	Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Memory Management Hardware	**5**
4.2	Multiprocessors: Characteristics of multiprocessor, Interconnection Structure, Interprocessor Communication & Synchronization	**5**

List of Practicals:

1. Design and realize all basic gates and universal gates.
2. Design and realize all basic gates from NOR and NAND gates.
3. Realize Half Adder, Full Adder, Half subtractor, Full subtractor
4. Realize a BCD adder
5. Realize a Serial Adder
6. Realize a four bit ALU
7. Realize Master-Save J K Flip-Flop, using NAND/NOR gates
8. Realize Universal Shift Register
9. Realize Self-Starting, Self Correcting Ring Counter
10. Realize Multiplexer and Demultiplexer
11. Realize Carry Look ahead Adder / Priority Encoder
12. Simulation of PAL and PLA

Learning Approach:

To understand the basics of Computer Architecture and Organization and their implementations and applications.

References/suggested learning resources:

a) Books

1. Morris Mano, Computer System Architecture, 3rd Edition, Prentice-Hall of India Private Limited, 1999.
2. WIliam Stallings, Computer Organization and Architecture, 4th Edition, Prentice Hall of India Private Limited, 2001

3. Subrata Ghosal," Computer Architecture and Organization" , Pearson 2011
4. Malvino, "Digital Computer Electronics: An Introduction to Microcomputers", McGraw Hill

b) Open source software and website address:

1. Electronics Workbench
2. MultiSim

BCA- DC103 | Problem Solving & Programming in C

Teaching Scheme			
Lecture hours per week	**Tutorial Hours per week**	**Practical hours per week**	**Credit**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

After studying this course, students will be able to:

1. Design a solution to the problem using an Algorithm, Flowchart, and Pseudocode.
2. Use operators and programming constructs in C.
3. Design & Develop functions for the problems.
4. Apply the concept of storage class, arrays, strings, and pointers including dynamic memory allocation, pre-processor directives, structures, union, enumeration, and command-line arguments.
5. Execute the file handling operations in C.

Syllabus:

Unit	Title	Hours
Unit 1	**Problem Solving and Program Design**	**10**
1.1	Problem Solving and Introduction to Programming: Activities involved in Problem Solving, Programming Language, Programming Paradigms, Characteristics of a good program, Classification of Programming Languages, Common programming errors.	
1.2	Program Design: Algorithm, Flowchart, Pseudocode: Introduction, Steps involved and detailed construction, Problems on selection, looping, flow chart analysis.	7
UNIT 2	**C Basics, Decision Making and Looping**	**10**

2.1	C basics: Steps involved in execution of a program, C character set, Identifiers and keywords, Data types, constants, variables and arrays, declarations, expressions statements, symbolic constants, compound statements, arithmetic operators, unary operators, relational and logical operators, assignment operators, conditional operators, bit operators, Arithmetic Expressions and its evaluation, Operator precedence and Associativity, Managing Input and Output, Formatted Input and Output.	
2.2	C constructs and flow charts: If statement, if…else statement, Nesting of if...else statement, The else if Ladder, while statement, do….while statement, for statement, switch statement, nested control statement, break operator, continue operator, comma operator, goto statement, flowchart for each construct.	
UNIT 3	**C Functions, Storage Classes, Arrays and String Handling**	**10**
3.1	C Functions: Built-in and User Defined Functions, Elements of User Defined Functions, parameter passing mechanisms and returning values, scope, recursion, nesting of functions.	
3.2	Arrays: One-Dimensional, Two-Dimensional and introduction to Multi-Dimensional Array, flow chart, and problem solving on array manipulation.	
3.3	Storage Classes: Automatic, external (global), static & registers.	
3.4	String Handling and library functions: String manipulation functions and other standard library functions	
UNIT 4	**Pointers, Pre-processor Directives, Structures and Basic File Management**	**10**
4.1	Pointers: Introduction, Chain of pointers, array & pointer relationship, pointer arithmetic, dynamic memory allocation, pointer to arrays, array of pointers, function returning pointers, pointers to functions, array of pointers to functions.	
4.2	Pre-processor: #include, #define, macros with arguments, the operators # and ##, conditional compilations	
4.3	Structures: Introduction to Structures, Unions and Enumerations, passing structure to functions, Pointer to Structures, bit fields	

4.4	Introduction to File handling [text (ASCII), binary]: Opening, closing, input/output, random access to files. Usage of command-line arguments.	

List of Practicals:

1. Programs on reading and displaying operands of different types, formatted input and output.
2. Programs on ASCII value of characters.
3. Programs on performing C operations on operands of different data types.
4. Programs on evaluation of arithmetic expressions and operator precedence.
5. Programs on decision making statements and branching: if, if..else, nesting of if..else, The else if Ladder, conditional operator, switch statement, goto, break, continue.
6. Create a user defined function, parameter passing & returning values.
7. Programs on recursion e.g. fibonacci series, factorial etc.
8. Programs on one-dimensional arrays and operations like sum, average etc.
9. Programs on matrices and operations on matrices.
10. Programs on multi-dimensional arrays.
11. Programs on storage classes.
12. Programs on string manipulation.
13. Program to print address of variable & its value.
14. Programs on pointer declaration, initialization & accessing the value pointed to by a pointer, use of pointers in arithmetic expressions.
15. Programs to perform array (single-dimensional & multi-dimensional - up to three dimensions) operations using pointers, array of pointers, function returning pointers & pointers to functions.
16. Programs on dynamic memory allocation.
17. Programs on preprocessor directives.
18. Programs on structures, unions, and enumerations, passing structures to functions, pointer to structures.
19. Programs on File Management.

Learning Approach:

To understand the basics of procedure-oriented programming and its applications.

References/suggested learning resources:

a) Books

1. Programming in ANSI C, E.Balagurusamy
2. Computer Basics and C Programming, V. Rajaraman
3. The Art of Programming Through Flowcharts & Algorithms, Anil Bikas Chaudhuri.

b) Open source software and website address:

1. Code::Blocks: https://www.codeblocks.org/
2. Visual Studio Code: https://code.visualstudio.com/
3. Flowgorithm: http://www.flowgorithm.org/

BCA- DC104 | Information Technologies

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

After studying this course, students will be able to:

1. Describe the major components of information technology applications:
2. Understand the basics of hardware, computer networks, software, data, processes, and people.
3. Describe the different components of a computer network.
4. Demonstrate an understanding of different types of networks.
5. Define "Software Engineering" & Software development Procedure .
6. Discuss the role of databases in IT applications.

Syllabus:

Unit	Title	Hours
Unit 1	**Introduction to Digital Electronics & System Processing**	**10**
1.1	Introduction to Digital Electronics: Digital Signals and Logic gates, Number systems: Binary, octal and hexadecimal number systems, signed binary number, binary arithmetic	
1.2	Information System Processing: Information system and information processing, Introduction to Database Management System and its types.	
UNIT 2	**Introduction of Information Security and Computer Graphics**	**10**

2.1	Introduction to Information Security, Security Goals, Attacks, Security Services and Mechanisms, Cloud Computing.	
2.2	Introduction of Computer Graphics: Display Devices: Refresh Cathode Ray Tube, Raster Scan Display, colour display techniques, interactive input/output devices. Animation Applications, Digital Image Processing, Multimedia and Applications.	
Unit 3	**Introduction to Operating System and Office Suite**	**10**
3.1	Operating System: Introduction to Operating system, Different types of operating systems and their working, DOS/ Linux commands, File Structure and Storage, Introduction to process management, Windows & Linux Operating System.	
3.2	Office Suite: Introduction, Microsoft Word, Microsoft Excel, Microsoft Powerpoint, OpenOffice	
Unit 4	**Introduction to Computer Network**	**10**
4.1	Computer Network: Basic Elements of a Communication System, Data transmission media, Digital and Analog Transmission, Network topologies, Network Types (LAN, WAN, and MAN), Introduction to Communication protocols, OSI Model, Internet, IOT.	

List of Practicals:

1. Program Based on Ms-Word.
2. Program Based on Ms-Excel.
3. Program Based on Ms-Powerpoint.
4. Developing Flow Chart.
5. Basic OS Command.
6. Basic Networking.

Learning Approach:

To understand the basics of Computer and Information Technology and their applications

References/suggested learning resources:

1. Alex Leon and Mathews Leon, “Fundamentals of Information Technology”, Leon Techworld, 2007.
2. P. K. Sinha and Priti Sinha , “Computer Fundamentals”, BPB Publications, 2007.
3. Malvino and Leach, “Digital Principles and Application”, TMH, 1999.

4. Alex Leon and Mathews Leon, “Introduction to Computers”, Vikas Publishing House, 2007.
5. Norton Peter, “Introduction to computers”, TMH, 4th Ed., 2006.
6. Simon Haykins, “Communication System”, John Wiley & Sons, 2006.
7. B. Basaraj, “Digital Fundamentals”, Vikas Publications, 1999.
8. V. Rajaraman, “Introduction to Information Technology”, PHI, 2006.

BCA- DC105 | Software Engineering

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

After studying this course, students will be able to:

1. To understand the iterative implementation of software projects.
2. To analyze projects using use case modelling tools.
3. To develop solutions for real-life cases using design models and patterns.
4. To understand and implement project design requirements for user interface, data layer, and system controls.
5. To apply modern case tools to develop solutions.

Syllabus:

Unit	Title	Hours
Unit 1	**Introduction, Process Model, Requirement Analysis & Specification**	**10**
1.1	Introduction: Software Crisis, Software Processes & Characteristics, Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models	
1.2	Software Requirements analysis & specifications: Requirement engineering, requirement elicitation techniques like FAST, QFD, Requirements analysis using DFD(with case studies), Data dictionaries & ER Diagrams, Requirements documentation, Nature of SRS, Characteristics & organization of SRS.	
Unit 2	**Software Project Management**	**10**

2.1	Software Project Management Concepts: The Management spectrum, Key Objectives of Effective Management	
2.2	Software Project Planning: Size Estimation like lines of Code & Function Count, Cost Estimation Models, COCOMO, Risk Management.	
Unit 3	**Software Design & Metrics**	**10**
3.1	Software Design: Cohesion & Coupling, Classification of Cohesiveness & Coupling, Layered arrangement of modules, Function Oriented Design, Object Oriented Design	
3.2	Software Metrics: Software measurements: What & Why, Token Count, Halstead Software Science Measures, Design Metrics, Data Structure Metrics.	
Unit 4	**Software Testing, Reliability & Maintenance**	**10**
4.1	Software Testing: Code Review, Testing Process, Types of Testing, Functional Testing, Structural Testing, Test Activities, Unit Testing, Integration Testing and System Testing(Performance Testing and Error Seeding), Debugging Activities.	
4.2	Software Reliability: Failure and Faults, Reliability Models: Basic Model software Quality Standards SEI & CMM Software Maintenance: Management of Maintenance, Maintenance Process, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation	

List of Practical:

1. Prepare Problem Statement
2. Prepare Feasibility Report
3. Prepare DFD, ERD
4. Prepare Use Case Diagram
5. Prepare Design Document
6. Prepare Structure Chart, Decision Table
7. Prepare Test Case & Test Suit
8. Prepare Test Report

Learning Approach:

To understand the basics of software engineering and its applications.

References/suggested learning resources:

1. K. K. Aggarwal & Yogesh Singh, “Software Engineering”, 2nd Ed., New Age International, 2005.
2. Rajib Mall, “Fundamental of Software Engineering”, 3rd Edition, PHI Learning Private Limited
3. I. Sommerville, “Software Engineering”, 9th Edition, Pearson Edu.
4. Jalote Pankaj , “Software Engineering”, Narosa Publication
5. R. S. Pressman, “Software Engineering – A practitioner’s approach”, 5th Ed., McGraw Hill Int. Ed., 2001.
6. James Peter, W. Pedrycz, “Software Engineering: An Engineering Approach”, John Wiley & Sons.

SEMESTER II

BCA- DC201 | Discrete Mathematics

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
3	1	0	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes

At the end of the course student should be able:

1. To get the Knowledge about sets, relations and functions.
2. To study the basics of lattices and graphs.
3. To get familiar with propositional logic.

Syllabus:

Unit	Title	Hours
Unit 1	**Set Theory, Relations & Functions**	**10**
1.1	Sets, Definitions, Types of Sets, Operations on Sets, Multisets, Computer Representation of Sets.	3
1.2	Types of Relations, Composition, Inverse, Closure, Matrix Representation of Relation, Partial Ordering, Relational Model for Databases.	4
1.3	Functions, Types of Functions, Composition, Inverse Hash Function, Growth of Function	3
Unit 2	**Mathematical Logic**	**10**
2.1	Introduction, Propositions statements, Truth Tables, Logic Equivalence, Algebra of Propositions, Types of Propositions, Tautologies & Contradictions	4
2.2	Normal forms, Logic in proof, Methods of Proof, Mathematical Induction, Predicate Calculus.	4
2.3	Number Theory, Divisibility theory, Congruences, Application of	2

	Congruences	
Unit 3	**Boolean Algebra & Logic Circuit**	**10**
3.1	Introduction, Boolean Algebra, Basic Operations, Boolean Function, DeMorgan’s Theorem, SOP & POS Form, Normal Form, Boolean Function as a Canonical form, Simplification of Boolean Expression by Algebraic Method, K-Maps.	5
3.2	Group Theory, Definition, Types of Groups, Homomorphism & Isomorphism of Groups.	5
Unit 4	**Graph Theory and Automata**	**10**
4.1	Graph Theory, Types of Graphs, Operation on Graphs.	3
4.2	Language Grammar & Automata: Introduction, Strings, Languages, Regular Expression, Grammars, Finite State Machines, Finite State Automata, Moore & Mealy Machine, Pushdown Automata, Turing Machine	7

Learning Approach:

To understand the basics of Discrete Mathematics and applications in Computer Science.

References/suggested learning resources:

1. A TextBook Of Discrete Mathematics : C.L. Liu
2. Discrete Mathematical Structures : Trembely and Manohar
3. A TextBook of Discrete Mathematics : Swapan Kumar Sarkar
4. Discrete Mathematics : S.K. Chakraborty and B.K.Sarka

BCA- DC202 | Operating Systems

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
3	1	0	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

Upon successful completion of this course, the student shall be able to:

1. Help students become familiar with the fundamental concepts of operating systems.
2. Help students become competent in recognizing operating systems features and issues.
3. Provide students with sufficient understanding of operating system design and how it impacts application systems design and performance.
4. Exhibit familiarity with the fundamental concepts of operating systems.
5. Exhibit competence in recognizing operating systems features and issues.
6. Apply a mature understanding of operating system design and how it impacts application systems design and performance.

Syllabus:

Unit	Title	Hours
Unit 1	**Introduction to Operating System, Processes & CPU Scheduling**	**10**
1.1	Introduction to Operating System: Introduction, Role, Types of OS; Batch Systems, multiprogramming, time-sharing, parallel, distributed and real-time systems, Operating system structure, Operating system components, and services.	
1.2	Processes: Process Concept, Process Scheduling, Operation on Processes, Cooperating Processes, Threads.	

1.3	CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple- Processor Scheduling.	
Unit 2	**Inter process Communication, Synchronization & Dead Locks**	**10**
2.1	Inter process Communication and Synchronization: Background, The Critical-Section Problem, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Critical Regions, Monitors, Message Passing.	
2.2	Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Deadlock Handling	
Unit 3	**Memory Management & Virtual Memory**	**12**
3.1	Memory Management: Background, Logical vs. Physical Address space, swapping, Contiguous allocation, Paging, Segmentation, Segmentation with Paging.	
3.2	Virtual Memory: Demand Paging, Page Replacement, Page-replacement Algorithms, Performance of Demand Paging, Allocation of Frames, Thrashing , Demand Segmentation	
Unit 4	**File-System Interface & Implementation**	**8**
4.1	File-System Interface: File Concept, Access Methods, Directory Structure.	
4.2	File-System Implementation: Introduction, File-System Structure, Basic File System, Allocation Methods, Free-Space Management, Directory Implementation.	

Learning Approach:

To understand the basics of operating systems and its applications.

References/suggested learning resources:

1. Silberschatz and Galvin, “Operating System Concepts”, John Wiley, 8th Ed., 2009.
2. Milan Kovic., “Operating Systems”, Tata McGraw Hill, 2001
3. Deitel, Deitel, and Choffnes, “Operating Systems”, Pearson,3rd Edition
4. Tannenbaum, “Operating Systems”, PHI, 4th Ed., 2000.
5. Madnick E. and Donovan J., “Operating Systems”, Tata McGraw Hill, 2001.
6. Flynn McHoes, “Operating System”, Cengage Learning, 2006.
7. Pbitra Pal Choudhury, “Operating System Principles and Design”, PHI, 2009.
8. William Stallings, “Operating Systems Internals & Design Principles”, Pearson

BCA- DC203 | Free & Open Source Software (Linux Programming & Administration)

Teaching Scheme			
Lecture hours per week	**Tutorial Hours per week**	**Practical hours per week**	**Credit**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

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

1. Explain the features of free & open source software
2. Familiarization with LINUX

Syllabus:

Unit	Title	Hours
Unit 1	**Open source software**	**8**
1.1	Features, advantages over proprietary software, examples, Free software: concepts, features, Free software Vs Open Source software, Free software movements. Policies, GPL, Free OS	4

1.2	Social Impact Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government.	4
Unit 2	**Introduction to Linux**	**8**
2.1	History and Features of Linux, Various flavours of Linux, Linux Kernel and Shell, Graphical Desktops- GNOME, KDE, Linux File System and Directories, Linux commands bc, cal, cat, cd, chgrp, chmod, clear, cmp, cp, kill, rm, rmdir, tty, wc, who, grep, write, telnet, whois, mv, find, ps, mkdir, more, date, mount, show, mount etc. Pipeline and redirection concepts, using floppy and cd-rom in linux	4
2.2	Shell Programming: Available shells under Linux (viz. Bash, TCSH, Korn or so on), different Shell features, editors, shell commands, shell scripts: shell variables, environmental variables, purpose of shell scripts, writing, storing and executing scripts, Filters- The grep family, advanced filters-sed and awk.	4
Unit 3	**Resource Management in Linux**	**8**
3.1	File and directory management, Command-line shortcuts, File Types, Ownership and Permissions, File management and manipulation, Moving users & its directories, Miscellaneous Tools, Editors, system calls for files Process Management, Signals, IPC: Pipes, FIFOs, System vs IPC, Message Queues, system calls for processes, Memory Management, library and system calls for memory.	5
3.2	Introduction to Networking in LINUX: Socket Introduction, Elementary TCP Sockets (Socket Function, Connect Function, Bind, Listen, Accept, Fork and Exec), TCP Client server Example, Elementary UDP Sockets.	3
Unit 4	**Linux Administration**	**8**
4.1	Managing Users and Groups Creating and managing user/s and group commands, User management Tools, Users and Access Permissions, Updating users and group attributes, PAM (Pluggable Authentication Modules)	4
4.2	Booting and Shutting down Boot Loaders, The init process, rc scripts, enabling and disabling services, Booting in recovery mode	4

List of Practicals:

1. Installation of Linux
2. Configuration of Linux

3. Shell Scripting Programs using various Looping, Read Statement, Formatted Output, Case Condition etc
4. Writing and Executing C & C++ Programs
5. Using Vi Editor & various Commands
6. Find, awk, grep commands
7. User Administration
8. Installation of Web Server

Learning Approach:

To understand the basics of open-source operating systems and their applications.

References/suggested learning resources

a) Books

1. Richard Peterson, Linux Programming: A Beginners Guide, DreamTech.
2. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006
3. Arnold Robbins, “Linux Programming by Examples: The Fundamentals”, Pearson Education, 2nd Ed., 2008.
4. Mark G. Sobell, “A Practical Guide to Ubuntu Linux”, Pearson, 2nd Ed., 2008.
5. Evi Nemeth, Garth Snyder, Trent R. Hein, “Linux Administrator Handbook”, Pearson, 2nd Ed., 2007.

b) Open source software and website address

1. The Linux Documentation Project: http://www.tldp.org/
2. Docker Project Home: http://www.docker.com
3. Linux kernel Home: http://kernel.org
4. Open Source Initiative: https://opensource.org/
5. Linux Documentation Project: http://www.tldp.org/
6. Wikipedia: https://en.wikipedia.org/
7. https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia
8. Github: https://help.github.com/
9. The Linux Foundation: http://www.linuxfoundation.org/

BCA- DC204 | Data & File Structure using C

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

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

1. Familiarize with the fundamentals of data and file structures and their operations like insertion, deletion, searching, and sorting.
2. Understand and implement data structures like arrays, linked lists, stacks, queues, trees, graphs, and heaps.
3. Identify suitable data structures to model data used in real-world applications.
4. Familiarize with File Structures & Hashing Techniques

Syllabus:

Unit	Title	Hours
Unit 1	**Introduction to Linear Data Structure: Array, Searching & Sorting Techniques**	**10**
1.1	Introduction to Data Structures: Basic Terminology, Elementary Data Organization, Classification of data structures and its operations, Abstract Data Type.	
1.2	Arrays, Sparse and Dense Matrices: Introduction to single and multidimensional arrays (up to three dimensions), Operations on Arrays, Memory Representation and address calculation, Introduction to Sparse Matrices and Dense Matrices, Sparse Matrices – Lower, Upper Triangular Matrices and Tridiagonal matrices; operations on Sparse Matrices.	
1.3	Searching: Introduction, Linear (Sequential) Search, and Binary Search.	
1.4	Sorting Techniques: Insertion Sort, Selection Sort, Bubble Sort,	

	Quick Sort and Merge Sort.	
UNIT 2	**Introduction to Linear Data Structures: Stacks, Queues and Linked List**	**10**
2.1	Linked List: Introduction to linked lists and representation in memory, Singly linked list and operations such as traversal, insertion, deletion, searching on it, Introduction to Circular linked list, Linked list with headers and Doubly Linked List, Applications of a Linked List. Polynomial Representation and Operations using Arrays & Linked List.	
2.2	Stacks & Queues: Introduction and primitive operations on stack, Arithmetic Expressions: Polish Notation, Stack application: Evaluation of postfix expression, Conversion from infix to postfix, Introduction and primitive operations on queues, Circular Queues, D-queues and Priority Queues, Application of Queues.	
Unit 3	**Introduction to Non-Linear Data Structures: Trees, Graphs and Heaps**	**10**
3.1	Trees: Introduction and terminology, Binary Trees, Representing Binary Trees in Memory, Linked Representation of Binary Tree, Traversal of binary trees, Recursive algorithms for tree operations such as traversal, insertion and deletion, Introduction to Expression Trees, Binary Search Trees, AVL trees, m-way search trees, Applications of Tree data structure.	
3.2	Graphs: Introduction and Terminology, Representation – Array Based, Linked and Set Representation, Graph Traversals and its applications.	
3.3	Heaps: Introduction, Structural Properties, Classification of Heap: Max-Heap Tree and Min-Heap Tree, Applications of Heap data structure.	
Unit 4	**Introduction to Hashing, File Structures and File Indexing Techniques**	**10**
4.1	Hashing: Introduction, Hash Table, Hash Functions: Types and Requirements, Collision Management – Chaining and Open Addressing, Hashing applications.	
4.2	File Structures: Introduction, Concept of Fields, Records and Files, Unordered File (Heap File) and Ordered File, Sequential File Organization, Direct File Organization, and Indexed Sequential Organization, Choice of File Organization.	
4.3	File Indexing Techniques: Introduction, Single-Level Ordered Indexes: Primary Index, Clustering Index, and Secondary Index, Introduction to Multilevel indexes, Indexing with binary search trees, B-Tree and B^{+}-Tree.	

List of Practicals:

The student is encouraged to use any programming language - C/C++/Java

1. Programs on arrays (single-dimensional and multidimensional - up to three dimensions), array operations such as traversing, insertion, deletion, merging, etc.
2. Programs on matrices, Triangular matrices, Diagonal matrices, and Tridiagonal matrices.
3. Programs to determine whether a matrix is Dense or Sparse matrix, operations on sparse matrices such as addition, subtraction, multiplication, and transpose.
4. Programs to perform linear search and binary search for an element from the entered numbers and determine its position using both arrays and linked lists.
5. Programs to implement different sorting techniques - Insertion Sort, Selection Sort, Bubble Sort, Quick Sort and Merge Sort.
6. Programs to implement a Linked List and perform operations on it such as Traversal, Searching, Determination of Predecessor and Successor of a node, Insertion, Deletion, Sorting, Merging, Reversing, Concatenation of Two Lists, Splitting of a Linked List.
7. Programs to implement Circular Linked List, Linked List with Header, and Doubly Linked List.
8. Programs on Polynomial Representation and Operations using arrays and linked lists.
9. Programs to implement Stack using array and linked list, Push and Pop operations on Stacks.
10. Program on Stack applications such as balanced parenthesis, evaluation of postfix expressions, and conversion of expressions from infix to postfix form.
11. Programs to implement Queue using array and linked list, Insertion & Deletion of the element from Queue.
12. Program to implement Circular Queue, D-Queue and Priority Queues.
13. Programs to implement a binary tree and perform the inorder, preorder and postorder traversal, insertion, and deletion of a node from the binary tree.
14. Program to implement a binary search tree and perform operations such as searching, insertion, and traversal.
15. Program to implement a graph, add or remove edges and vertices to the graph.
16. Program to construct a Heap given the Heap Elements.
17. Program to implement the Hash Function.
18. Program to prepare the Hashing Table, enter the elements and map them in the index using Hash Function.

Learning Approach:

To understand the basics of data structures and file structures and their applications.

References/suggested learning resources:

a) Books

1. Introduction to Data Structures in C, Ashok N. Kamthane, Pearson
2. Data Structures Using C, A.K.Sharma, Pearson
3. Data Structures With C, Seymour Lipschutz, McGraw Hill Education
4. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed.

b) **Open source software and website address:**

1. Code::Blocks: https://www.codeblocks.org/
2. Visual Studio Code: https://code.visualstudio.com/
3. Flowgorithm: http://www.flowgorithm.org/

BCA- DC205 | Object Oriented Programming using C++

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

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

1. Learn how to implement Object Oriented concepts through C++.
2. To gain knowledge of objects, Class, Data Abstraction, Encapsulation, and Inheritance.
3. To understand Polymorphism and Dynamic Binding.
4. To know about constructing programs using the Bottom-up design approach.

Syllabus:

Unit	Title	Hours
Unit 1	**Introduction to OOPS and C++ environment**	**10**
1.1	Introduction: Introducing Object-Oriented Approach, Relating to other paradigms (functional, data decomposition). Features of Procedure oriented programming, Basic Concepts of Object Oriented Programming, Benefits of OOP, Applications of OOP, Difference between C and C++, cin, cout, new, delete operators.	
1.2	C++ Environment: Program development environment, the language and the C++ language standards, C++ standard libraries. Introduction to various C++ compilers, C++ standard libraries, Testing the C++ program in Turbo C++/Borland, C++/Microsoft VC++/GNU C++ compiler.	
UNIT 2	**Classes and Objects**	**10**

2.1	Classes and Objects: Encapsulation, information hiding, abstract data types, Object & classes, attributes, methods, C++ class declaration, references, this pointer, Function Overloading, Constructors and destructors, instantiation of objects, Default parameter value, C++ garbage collection, Dynamic memory allocation, Meta class/abstract classes.	
Unit 3	**Inheritance and Polymorphism**	**10**
3.1	Inheritance and Polymorphism: Inheritance, Class hierarchy, derivation – public, private & protected, Aggregation, composition v/s classification hierarchies, Polymorphism, Categorization of polymorphism techniques, Method polymorphism, Polymorphism by parameter, Operator overloading, Parametric polymorphism, Virtual Function, Early v/s Late Binding.	
Unit 4	**Generic Programming and Files and Exception Handling**	**10**
4.1	Generic Programming – Introduction, templates, template functions, Overloading of template functions, Overriding inheritance methods.	
4.2	Files and Exception Handling: Persistent objects, Streams and files, Namespaces, The basic stream classes: C++ predefined streams, Error handling during file operations, Command Line Arguments. Types of Exception, Catching and Handling Exceptions.	

List of Practicals:

Practicals will be based on the topic covered in the content. Therefore students must understand the concept and implementation of the contents in the syllabus by doing practicals as suggested by the faculty member.

Learning Approach

To understand the basics of Object-Oriented concepts and their implementation.

References/suggested learning resources:

a) Books

1. Ashok N. Kamthane, “Object-Oriented Programming With Ansi And Turbo C++”, Pearson Education.
2. A.R.Venugopal, Rajkumar, T. Ravishanker “Mastering C++”, TMH, 1997.
3. E. Balguruswamy, “C++ ”, TMH Publication ISBN 0-07-462038-x.
4. Mahesh Bhave, “Object-Oriented Programming with C++”, Pearson Education.
5. D . Parasons, “Object-Oriented Programming with C++”, BPB Publication. 6. Steven C. Lawlor, “The Art of Programming Computer Science with C++”, Vikas Publication.
7. Schildt Herbert, “C++: The Complete Reference”, 4th Ed., Tata McGraw Hill, 1999.

8. R. Lafore, “Object-Oriented Programming using C++”, Galgotia Publications, 2004.

b) Open source software and website address

1. TURBO C / Code::Blocks
2. TURBO C++
3. https://www.udemy.com
4. https://www.coursera.org

SEMESTER III

BCA- DC301 | Web Technologies & E-Commerce

Teaching Scheme			
Lecture hours per week	**Tutorial Hours per week**	**Practical hours per week**	**Credit**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

After studying this course, students will be able to:

In this course, the learners will be able to develop expertise related to the following:

- Students should be able to design and implement a basic website.
- Students should be able to implement different navigation strategies.
- Students should be able to develop a simple back-end database to support a website.
- Students should be able to recognize and evaluate website organizational structure and design elements.

Syllabus:

Unit		Hours
1	**HTML BASICS AND SERVERS**	**12**
1.1	History of the Internet and World Wide Web, Search Engines, News-group, E-mail and its Protocols, Web Portal, Browsers and their versions, Its functions, URLs, web sites, Domain names, Portals. Introduction to web applications, HTML, Client Side Scripting Vs Server Side Scripting,	
1.2	Web Servers : Local Servers and Remote Servers, Installing Web servers, Internet Information Server (IIS) and Personal Web Server (PWS)	
1.3	Static Web Development: HTML - Introduction to HTML, HTML Document structure tags, HTML comments, Text formatting, inserting special characters, anchor tag, adding images and Sound, lists types of lists, tables, frames and floating frames, Developing Forms, Image maps. Static website vs Dynamic website development. Cascading Style Sheet, Introduction to JavaScript.	
2	**PHP-ARRAYS AND FILE**	**10**
2.1	Introduction to PHP, Start and End Tags of PHP, Data types in PHP, Variables, Constants, operators and Expressions, printing data on PHP page, Control statements – if, switch case, for, while, do while.	
2.2	Arrays: Initialization of an array, Iterating through an array, Sorting arrays, Array Functions, Functions: Defining and Calling Functions, Passing by Value and passing By references, Inbuilt Functions.	
3	**FORMS AND SESSION**	**10**
3.1	Working with Forms: Get and Post Methods, Query strings, HTML form controls and PHP	
3.2	Maintaining User State: Cookies, Sessions, and Application State.	
3.3	Working With Files: Opening and Closing Files, Reading and Writing to Files, Getting Information on Files	

4	**DATABASE CONNECTIVITY**	8
4.1	PHP Database Connectivity: Introduction to MYSQL, Creating database and other operations on database, connecting to a database, Use a particular database, Sending query to database.	
4.2	Parsing of the query results, Checking data errors.	

List of Practicals:

Note: Practical may be covered as per the syllabus by the faculty.

Learning Outcomes:

To understand HTML, PHP and Database connectivity with PHP.

References/suggested learning resources:

Books

Textbooks:

- The complete reference HTML, by Thomas A powell, TMH publication.
- Mastering HTML 4.0 by Deborah S. Ray and Erich J. Ray. BPB Publication.
- Programming PHP. Rasmus Lerdorf, Kevin Tatroe. (O'Reilly, ISBN 1565926102).
- PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Websites by Robin Nixon O'Reilly Media; 1 edition

Reference books:

- Core PHP Programming. Leon Atkinson (Prentice Hall, ISBN 0130463469).
- Beginning PHP5 and MySQL: From Novice to Professional, W. Jason Gilmore, 2004, Apress, ISBN: 1-893115-51-8

BCA- DC302 | Computer Graphics

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

After studying this course, students will be able to:

In this course, the learners will be able to develop expertise related to the following: -

- Basic building blocks and core concepts of computer graphics.
- To understand Basic Graphics Primitives.
- Theoretical, mathematical foundation, and practical aspects of different graphics algorithms.
- Fundamental concepts of animation and its related technologies to design interactive graphics applications.

Syllabus:

Unit	Title	Hours
1	**Introduction of Graphics**	**10**
1.1	Introduction: Computer Graphics and its Applications, Overview of Graphics Systems–Video display devices, Raster-Scan Systems, Random-Scan Systems, Graphics Software.	
1.2	Scan Conversion: Scan conversion of Point and Line, Line Drawing Algorithms: DDA, Bresenham’s, Circle: Mid-Point, Bresenham’s,	
1.3	Animation: Introduction to Animation, Principles, Animation Methods, Animation Tools	
2	**Transformation**	**8**

2.1	Transformations: Two-Dimensional and Three-Dimensional Geometric Transformations: Translation, Scaling, Rotation, Reflection, Shearing, Homogeneous Coordinates, Composite Transformations, Rotation about an arbitrary point, Reflection about an Arbitrary Line, Affine Transformations	
2.2	Window to View port: Windowing, 2D-Viewing Pipeline, Computing Location of Viewport, Window to Viewport Transformation	
3	**Clipping , Projection and Solid Modelling**	**12**
3.1	Clipping: Introduction, Point Clipping, Line Clipping, Line Clipping Algorithms: Cohen Sutherland Algorithm, Polygon Clipping, Polygon Clipping Algorithms: Sutherland Hodgeman Polygon Clipping	
3.2	Projections: Introduction to Projections, Types of Projections: Parallel and Perspective Projections, Parallel Projections: Oblique, Orthographic, Axonometric, Perspective Projections: One-Point, Two-Point, Three-Point, Vanishing Points, Perspective Anomalies, Comparison of Parallel and Perspective Projections	
3.3	Solid Modelling: Solid Representation, Sweep Representation, Boundary Representation (B-Rep), Octrees, Constructive Solid Geometry (CSG), Comparison of Representations	
4	**Curves and Surfaces, Hidden Surface Removal**	**10**
4.1	Curves and Surfaces: Curves Representation, Parametric and Non-Parametric Curves, Bezier Curves: Cubic and Higher-Order Curves, Blending Function, Bernstein Polynomial, Continuity Conditions, Limitations, B-Splines: Construction of B-Spline Curves, Cubic B-Spline Curves, Knot Vectors, Uniform, Open-Uniform, Non-Uniform, Non-Uniform Rational Basis Splines (NURBS), Introduction to Bezier and B-Spline Surfaces.	
4.2	Hidden Surface Removal: Need for Hidden Surface Removal, Categories of Hidden Surface Removal Methods, Z-buffer, and A-buffer, Depth Sorting, BSP-Tree, Back-Face Removal, Scan-Line Method, Area-Subdivision Method.	

List of Practicals:

Practical Based on Syllabus (May be Implemented in C/ C++ Programming Language)

- WAP to Draw Line Using DDA Algorithm
- WAP to Draw Line Using Brasenham's Algorithm

- WAP to Draw Circle Using DDA Algorithm
- WAP to Draw Circle Using Brasenham's Algorithm
- WAP to Translate an Object
- WAP to Rotate an Object about 60 degrees in clockwise direction
- WAP to Scale an Object to twice its size
- WAP to Clip an Object

Note: Additional Practical may be covered as per the syllabus by the faculty

Learning Approach:

In this course, the learners will be able to develop expertise related to the following -

1. Demonstrate graphics drawings with respect to Graphics Primitives.
2. Apply 2D & 3D transformation concepts to represent images with different dimensions
3. Analyse and evaluate the concepts of projection and shading methods to obtain scenes with different clipping

To Understand Fundamental concepts of animation and its related technologies to design interactive graphics applications.

References/suggested learning resources::

Books

Textbooks:

1. Donald Hearn and Pauline Baker, "Computer Graphics with C Version ", Pearson, 2ndEdition,2002.
2. Foley James D, "Computer Graphics: Principles and Practice", Addison-Wesley Professional,3rdEdition,2013.
3. Rogers and Adams, "Mathematical Elements for Computer Graphics", McGraw-Hill, 2ndEdition, 2002.

Reference books:

1. John M. Blain, "The Complete Guide to Blender Graphics: Computer Modelling & Animation", CRC Press, 5thEdition,2019.
2. Malay K. Pakhira, "Computer Graphics Multimedia Animation", PHI,2ndEdition,2010.
3. F. S. Hill, Jr. Stephen M .Kelley, "Computer Graphics using OpenGL", Pearson,3rdEdition,2008.
4. Rajesh K Maurya, "Computer Graphics with Virtual Reality System", Wiley, 2ndEdition, 2014.
5. Rogers, "Procedural Element of Computer Graphics", McGraw-Hill,2ndEdition,2001.
6. Zhigang Xiang and Roy Plastock, "Computer Graphics Schaum's Outlines Series", McGraw-Hill,2ndEdition,2015.

BCA- DC303 | Python Programming

Teaching Scheme			
Lecture hours per week	**Tutorial Hours per week**	**Practical hours per week**	**Credit**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

- Design a solution to the problem using an Algorithm, Flowchart, and Pseudocode.
- Use python programming constructs, arrays, strings, functions, and modules to solve real-world problems.
- Design & develop custom functions & modules.
- Apply data structure primitives like lists, tuples, sets, and dictionaries.
- Apply file management operations to operate on files.
- Design and develop object-oriented solutions to problems using python programming language.
- Develop GUI & networking applications using python.
- Develop python applications with database connectivity operations.
- Familiarize and apply basic data science operations using Numpy, Pandas and Matplotlib library.

Syllabus:

Unit	Title	Hours
Unit 1	**Principles of Programming Language and Python Preliminaries**	**10**
1.1	**Concepts of Programming Language:** Introduction; Programming Domains, Language Evaluation Criteria, Influences on language design, Programming Paradigms, Implementation Methods: Compilation, Pure Interpretation, and Hybrid Implementation System.	
1.2	**Revisit to Program Design:** Algorithm, Flowchart & Pseudocode: Steps involved, Detailed Construction, Problems on Selection, Looping and Flow chart analysis.	

1.3	**Introduction to Python:** Features of Python, Structure of Python Program, Interactive Execution, Installation and working on Python IDE: Jupyter/PyCharm/IntelliJ IDEA and others, Identifiers, Keywords, Delimiters and Literals, Statements, Variables, Escape Sequences; Comments, Data Types, Type Conversion Functions and Rounding, Operators: Arithmetic, Assignment, Unary Minus, Relational, Logical, Boolean, Bitwise, Membership: in, not in, Identity: is, is not; Expressions, Operator Precedence and Associativity, Input and Output Statements, Command Line Arguments, Short-circuit, and Lazy Evaluation.	
1.4	**Control Flow Statements:** Conditionals: if Statement, if-else Statement, Conditional Expressions; Nested Conditionals: Nested if and Multi-Way if-elif-else Statements; Looping: while statement, for statement, else suite, The range() Function; Nested loops; The break statement; The continue Statement; The pass Statement; The assert statement; The return Statement.	
Unit 2	**Functions, Modules and Built-in Primitive Data Structures in Python**	**10**
2.1	**Functions:** Built-in, User-Defined and Anonymous function; Elements of User-Defined Functions, Arguments and Return Values, Formal vs. Actual Arguments, Scope and Lifetime, Positional, Keyword Arguments & Default Arguments, Nested Functions, Using Lambdas with filter(), map() and reduce() function, Decorators, Iterators, Generators, Recursion.	
2.2	**Modules:** Importing Modules; Math, Random and other standard library modules, Packages, Custom Modules.	
2.3	**Built-in Primitive Data Structures** **Arrays:** Single dimensional, Multi-dimensional arrays (up to three dimensions), Array Creation using array, linspace, logspace, arrange, zeros, ones, Operations on Arrays. **Strings:** Basics, Immutability, String creation, String Indexing and Slicing, String Manipulation, The subscript operator, Searching substrings.	
Unit 3	**File Management and Object Oriented Programming**	**10**
3.1	**File Handling:** Text and Binary Files - Writing and Reading Operations, Random Access to Files, The with Statement, Pickle in Python, Manipulating Files and Directories, Closing Files.	
3.2	**Introduction to Object-Oriented Programming:** Features, Classes & Objects, Immutable vs. Mutable Objects, Access Modifiers, Attributes and methods, Data Hiding, The 'self' variable, Constructor, Instance Variables and Class or Static Variables, Inner Classes, Passing members of one Class to another Class	

3.3	**Exception Handling: Error, Exception:** Preliminaries and Exception Class Hierarchy, Handling Exception using try, except and finally clauses, Raising Exceptions, Assertions, User-Defined Exceptions, Exception logging.	
Unit 4	**Advanced Python**	**10**
4.1	**GUI Programming with Tkinter:** Creating User-interface; GUI Widgets, Creating Layouts, Check Box, Radio Buttons, List Box, Menus, Menus Options, Dialog Boxes, Tables.	
4.2	**Network Programming:** Basics of Sockets, Socket Methods; TCP and UDP Sockets, Two-way Client-Server Communication, Sending email.	
4.3	**Database Access:** Advantages of a DBMS over Files, Database Connectivity Operations: Create, Insert, Select, Delete, Drop, Update & Joins.	

List of Practicals:

1. Exercise to set up Python Environment and IDE - Jupyter Notebook/ PyCharm/ IntelliJ IDEA/ other.
2. Exercises on variables and assigning values, type conversion.
3. Exercises on input, output and formatted output.
4. Exercises on Python operators, Escape Sequences and Comments.
5. Exercises on Operator Precedence and Associativity.
6. Exercises on number system conversion.
7. Exercises on expression evaluation.
8. Exercises on command line arguments.
9. Exercises on if statement, if...else statement, if...elif...else statement, conditional expressions, Nested Conditionals.
10. Exercises on looping, range() function, nested loops, the else suite.
11. Exercises on break statement, continue statement, pass statement, assert statement & return Statement.
12. Exercises on built-in functions.
13. Exercises on user-defined functions, passing & returning values, variable-length arguments.
14. Exercises on lambda functions, using lambdas with map(), reduce(), and filter function.
15. Exercises on Function Decorators, Generators & Iterators.
16. Exercises on nested functions & recursion.
17. Exercises on Importing python modules, using defined functions, and creating custom modules.
18. Exercises on array creation, accessing the elements & processing/manipulating them (single-dimensional & multidimensional up to three dimensions).
19. Exercises on string creation, string operations such as determination of length, indexing & slicing, the subscript operator, repeating the strings, concatenation, checking for substring & obtaining the position, comparing the strings, removing spaces, replacing substring, splitting, joining, checking starting & ending of a string, string & character testing methods, formatting strings, searching, sorting, insert substring into string.
20. Exercises on List creation and manipulation, List methods, List Comprehension, List Cloning, Searching and Sorting Lists, Tuple creation, accessing elements & operations, List to Tuple conversion & vice-versa.

Note: Additional Practical may be covered as per the syllabus by the faculty.

Learning Approach:

1. Design solution to the problem.
2. Master the fundamentals of writing Python scripts.
3. Use control constructs, functions & modules.
4. Develop custom functions & modules.
5. Use data structures in python.
6. Apply file handling operations and object-oriented programming techniques.
7. Explore GUI programming, network programming, and database operations in python.
8. Apply basic data science operations.

References/suggested learning resources

(a) Books

1. Concepts Of Programming Languages, Sebesta, Pearson Addison Wesley.
2. Programming languages: Design and implementation, Terrance W. Pratt, Marvin V. Zelkowitz, T.V. Gopal
3. The Art of Programming Through Flowcharts & Algorithms, Anil Bikas Chaudhuri.
4. Beginning Python From Novice to Professional, Magnus Lie Hetland, Apress
5. Python for Data Analysis, Wes McKinney, O'Reilly Media, Incorporated
6. Python Data Analytics With Pandas, NumPy, and Matplotlib, Fabio Nelli, Apress
7. Exploring Python, Timothy A. Budd, McGraw-Hill Education
8. Learning Python, Mark Lutz, O'Reilly Media, Inc.
9. Introduction to Programming Using Python, Liang Y. Daniel, Pearson Education
10. Programming in Python 3: A Complete Introduction to the Python Language, Mark Summerfield (Author), Pearson Addison-Wesley Professional.
11. Core Python Programming, R. Nageswara Rao, Dreamtech Press

BCA- DC304 | Database Management Systems

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcomes:

- The concepts related to database, database techniques, SQL and database operations are introduced in this subject. This creates a strong foundation for application data design.
- Explain the various database components, models, DBMS architecture and Database Security
- Construct advanced SQL queries on data and apply Procedural abilities through PL/SQL.
- Examine the use of normalization and functional dependency for database design.
- Appraise the concepts of transaction, concurrency control and recovery in databases.

Syllabus:

Unit	Title	Hours
Unit 1	**Introductory Concepts of DBMS**	**10**
1.1	Basic concepts: Database & database users, characteristics of the databases, database systems, concepts and architecture, Data Models, Schemas & instances, DBMS architecture & data independence, Overview of hierarchical, Network & Relational Database Management Systems.	4
1.2	Data Modelling using the Entity-Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, Concepts of keys, Extended ER model - Generalization, Specialization, Aggregation, ER diagram to tables Mapping.	6
UNIT 2	**Data Models and SQL**	**10**

2.1	Data Models: The importance of data models, Basic building blocks, Business rules, The evolution of data models, Degrees of data abstraction, Logical view of data, keys, integrity rules.	4
2.2	Relational Model: Relational data model, Relational integrity constraints: Entity Integrity, Referential integrity, Domain Constraints, Key constraints. Relational Algebra Introduction on SQL: SQL commands and types, DML, DDL, DCL, TCL. SQL Data Types and literals, Operators in SQL. Database Objects: Table, View, Sequence, index, Synonym, Queries	6
UNIT 3	**Normal Forms and PL/SQL**	**10**
3.1	Normal Forms: Functional Dependencies, Non-loss Decomposition, First, Second, Third Normal Forms, Dependency Preservation, Boyce/Codd Normal Form, Multi-valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form.	5
3.2	PL/SQL Programming: Introduction to PL/SQL, Structure of PL/SQL Block, PL/SQL language: Operators, Control Structure, Cursors, Triggers, Procedures and functions.	5
UNIT 4	**Transaction Management**	**10**
	Transaction Management: ACID properties, serializability of Transaction, Testing for Serializability and concurrency control, Lock-based concurrency control (2PL, Deadlocks), Time stamping methods, Database recovery management, Recovery Techniques.	10

List of Practicals

1. Design a Database and create required tables. For e.g. Bank, College Database
2. Apply the constraints like Primary Key, Foreign key, NOT NULL to the tables.
3. Write a SQL statement for implementing ALTER, UPDATE and DELETE
4. Write the queries to implement the joins
5. Write the queries for implementing the following functions: MAX, MIN, AVG, COUNT
6. Write the queries to implement the concept of Integrity constraints
7. Write the queries to create the views
8. Perform the queries for triggers
9. Perform the following operation for demonstrating the insertion, updation, and deletion using the referential integrity constraints

Note: Additional Practical may be covered as per the syllabus by the faculty

Learning Approach:

In this course, the learners will be able to develop expertise related to the following:-

- Develop a broad understanding of database concepts and Database Management System Software, data models, schemas and instances, data constraints, relational algebra, and calculus.
- Acquire Knowledge to model an application's data requirements using conceptual modelling tools like ER diagrams and design database schemas based on the conceptual model.

- Be able to write SQL and PL/SQL commands to create and manipulate database objects.
- Be able to discuss the importance of normalization and improve the database design by applying various normal forms.
- Get in-depth knowledge of concurrency control mechanisms, transaction management techniques, and database security.

References/Suggested learning resources:

(a) Books

Text Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudharshan, ―Database System Concepts‖, 5th Edition, Tata McGraw Hill, 2006
- Elmasri and Navathe, ―Fundamentals of Database Systems‖, 6th Ed., Pearson, 2013
- C.J.Date, A.Kannan, S.Swamynathan, ―An Introduction to Database Systems‖, 8th Edition, Pearson Education, 2006.
- J. D. Ullman, ―Principles of Database Systems‖, 2nd Ed, Galgotia Publications, 1999.
- Vipin C. Desai, ―An Introduction to Database Systems‖, West Publishing Co.

(b) Open source software and website address:

- Oracle 10g
- MySql

BCA- DC305 | Database Management Systems

Teaching Scheme			
Lecture hours per week	Tutorial Hours per week	Practical hours per week	Credit
2	1	2	4

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcome:

- Understand basics, topologies, and working mechanisms of wired and wireless computer networks.
- Analyze the features and operations of protocols of OSI reference model & TCP/IP protocol suite.
- Design, calculate, and apply routing mechanisms for IPv4 & IPv6
- Identify the networking requirements for an organization and select & propose appropriate architecture and technologies.
- Work on Network addressing, design and implementation.

Syllabus:

Unit	Title	Hours
Unit 1	**Basics of Computer Network**	**10**
1.1	Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, networks topology. Introduction to 5 Layer TCP/IP Network Model, Comparison with 7 Layer OSI Reference Model.	**5**
1.2	Physical Layer: The Physical Layer, Theoretical Basis for Data Communication, Guided Transmission Media, Wireless Transmission, Communication Satellites, Need for Modulation, Digital Signal Encoding Formats – NRZ-L, NRZI, bipolar-AMI, Manchester, Differential Manchester, Digital Modulation.	**5**
Unit 2	**Data Link Layer**	**10**

2.1	The Data Link Layer: Data Link Layer Design Issues, Error Detection and Correlation, Flow Control Protocols, Stop-and-wait Flow Control, Sliding – Window Flow Control, Error Control, Stop-and-wait ARQ, Go-back-N, Selective-repeat, Example of Data Link Protocols-HDLC.	5
2.2	Medium access sublayer: Channel allocations, ALOHA Protocols, Carrier Sense Multiple Access Protocols, CSMA with Collision Detection, Collision free protocols, Ethernet	5
Unit 3	**Network Layer**	**10**
3.1	Network Layer: Point-to-Point network, routing algorithms, congestion control, internetworking, Quality Control	5
3.2	The Network Layer in the Internet, IP packet, IP addresses, IPv4 Addressing, Subnetting & Supernetting, Need for NAT, Multicasting, Introduction to IPv6 Addressing, Need for IPv6 Addressing, Global, Local and Site Local Addressing, Multicast, Broadcast, Anycast and Unicast Addressing	5
Unit 4	**Transport & Application Layer**	**10**
4.1	Transport Layer: Design Issue, connection management, TCP window management, User Datagram Protocol, Transmission Control Protocol, Performance Issues, QOS, Transport Layer Security, IP Security, IPSec	5
4.2	Application Layer: DNS, Electronic Mail, WWW, Streaming Protocols, DHCP & DHCPv6, SMTP, Telnet, SSH, POP, POP3, IMAP Network Security: Need for Network Security, Cryptography and Compression Techniques, Firewall, Various types of Cyber Threats	5

List Of Practicals:

1. Study of Network devices in detail
2. Connect the computers in Local Area Network
3. Implementation of Data Link Framing method - Character Count.
4. Implementation of Data link framing method - Bit stuffing and Destuffing.
5. Implementation of Error detection method - even and odd parity.
6. Implementation of Error detection method - CRC Polynomials.
7. Implementation of Data Link protocols - Unrestricted simplex protocol
8. Implementation of data link protocols - Stop and Wait protocol
9. Implementation of routing algorithms - Dijkstra’s algorithm
10. Study of Network IP Addressing

Note: Additional Practical may be covered as per the syllabus by the faculty

Learning Approach:

1. Understand basics, topologies and working mechanism of wired and wireless computer networks.
2. Analyze the features and operations of protocols of OSI reference model & TCP/IP protocol suite.
3. Design, calculate, and apply routing mechanisms
4. for IPv4 & IPv6
5. Identify the networking requirements for an organization and select & propose appropriate architecture and technologies.
6. Work on Network addressing, design and implementation

References/suggested learning resources:

(a) Books

1. Forouzan, “Data Communication and Networking”, TMH, 4th Edition.
2. Behrouz A. Forouzan, “TCP/IP Protocol Suite”, Tata Mc Grew Hill, 4th Edition, 2009.
3. A.S. Tanenbaum, “Computer Networks”, PHI, 4th Edition.
4. Silvia Hagen, IPv6 Essentials. O'Reilly Media, Inc., 2006
5. W. Stallings, “Data and Computer Communication”, Macmillan Press.
6. Comer, “Computer Networks and Internet”, PHI. 5. Comer, “Internetworking with TCP/IP”, PHI
7. Kevin R Fall and W. Richard Stevens, “TCP/IP illustrated, Volume 1: The Protocols”, Addison Wesley, 2nd Edition, 2011.

(b) Open source software and website address:

1. https://onlinecourses.nptel.ac.in/noc21_cs18/preview
2. http://www.tcpipguide.com/
3. https://www.coursera.org/learn/managing-network-cybersecurity/home/welcome
4. https://www.coursera.org/learn/wireless-communications/home/welcome
5. https://www.coursera.org/learn/tcpip/home/welcome
6. https://futureskillsprime.edcast.com/pathways/introduction-to-cyber-security

SEMESTER IV

BCA-DC401 | Java Programming

Teaching Scheme			
Lecture hours per week	**Tutorial hours per week**	**Practical Hours per week**	**Credit**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcome:

- To make students well versed with programming in java.
- To understand the use of object oriented features along with their applications.

Syllabus:

Unit	Title	Hours
Unit 1	Basics of Java programming	10
1.1	Java Programming: History and Features of Java, C++ vs. Java, JDK, JRE, and JVM (Java Virtual Machine),JVM Memory Management, Internal Details of JVM,Introduction, Data types, access specifiers, operators, control statements, arrays, loops, Input-Output Stream. Classes: Fundamentals, objects, methods, constructors.	6
1.2	Inheritance: Super class, subclass, this and super operator, method overriding, use of final, packages, abstract class, interface. Polymorphism: Method overloading, constructor overloading.	4

UNIT 2	Exception Handling and Multithreading	10
2.1	Exception Handling: Exception Class, built in checked and unchecked exceptions, user defined exceptions, use of try, catch, throw, throws, finally.	3
2.2	Multi threaded programming: Overview, comparison with multiprocessing ,Thread class and runnable interface, life cycle, creation of single and multiple threads, thread priorities, overview of Synchronization.	3
2.3	Java Library: String handling (only main functions), String Buffer class. Elementary concepts of Input/Output: byte and character streams, System.in and Sysem.out, print and println, reading from a file and writing in a file.	4
UNIT 3	Applets and Event Handling	10
3.1	Applets :Introduction, Life cycle, creation and implementation, AWT controls: Button, Label, TextField, TextArea, Choice lists, list, scrollbars, check boxes, Layout managers,	4
3.2	Elementary concepts of Event Handling: Delegation Event Model, Event classes and listeners, Adapter classes, Inner classes.	3
3.3	Swings: Introduction and comparison with AWT controls.	3
UNIT 4	Networking and Servlets	10
4.1	Networking Basics: Socket (datagram and TCP/IP based client and server socket), factory methods, InetAddress	5

	JDBC: JDBC Architecture, JDBC Drivers, Connecting to the Database.	
4.2	Introduction to Java Servlets: Life cycle, Interfaces and classes in javax, servlet package (only description) Creating a simple servlet.	5

List of Practicals:

List of practicals will be prepared as per subject teacher.

Learning Approach:

Students will be well versed with programming in java and understood the use of object oriented features along with their applications.

References/Suggested Books :

Books

TEXT BOOKS:

- Patrick Naughton and Herbert Schildt, "Java-2 The Complete Reference", TMH.
- Y. Daniel Liang, "Introduction to Java Programming, Comprehensive Version, 7/e" Pearson

REFERENCE BOOKS: -

- Krishnamoorthy R, Prabhu S ,"Internet and Java Programming", New Age Intl.
- David Flanagan, Jim Farley, William Crawford and Kris Magnusson, "Java Enterprise in a Nutshell", O'Reilly.

BCA-DC402 | Design and Analysis Algorithms

Teaching Scheme			
Lecture hours per week	**Tutorial hours per week**	**Practical Hours per week**	**Credits**
2	**1**	**2**	**4**

Rationale:

The accelerated expansion of computing technologies and applications into all our lives means students need to to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

Learning Outcome:

- Understand different notions of asymptotic complexity
- Find the time and space complexity of an algorithm
- Understand various algorithms design techniques like greedy, divide and conquer, and dynamic programming.
- Understand the NP completeness.
- Implement, analyze, and compare algorithms.

Syllabus :

Unit	Title	Hours
Unit 1	Elementary Algorithmic	10
1.1	Introduction, Problems and instances, The efficiency of algorithms, Average and worst-case analyses, What is an elementary operation, why look for efficiency. Asymptotic Notation: Introduction, A notation for “the order of”, Other asymptotic notation, Conditional asymptotic notation, Conditional asymptotic notation, Asymptotic notation with several parameters, Operations on asymptotic notation.	5

1.2	Analysis of Algorithm: Introduction, Analyzing control structures, Using a barometer, Supplementary examples, Average-case analysis, Amortized analysis, Solving recurrences. Mathematical analysis of Recursive and Non-recursive algorithms.	5
UNIT 2	Greedy Algorithms	10
2.1	General characteristics of greedy algorithms; The knapsack problem; Minimum spanning trees: Prim, Kruskal algorithms, implementation issues and complexity analysis; Shortest path problem; scheduling	5
2.2	Introduction; Examples: Job sequencing with deadlines, Union-find data structure, Quick-union algorithm, Shortest path, Dijkstra's shortest path algorithm, Optimal merge. Huffmann Trees.	5
UNIT 3	Divide-and-conquer	10
3.1	Introduction; Finding the median and selection problem; Matrix Multiplication; Exponentiation	5
3.2	Merge sort, Quick sort, Binary Search, Binary tree Traversals and related properties, Multiplication of large integers.	5
UNIT 4	Dynamic Programming	10
4.1	Introduction and basic concepts; Calculation the binomial coefficient; The World Series; Making change;0-1 knapsack problem; All Pair Shortest paths and Transitive Closure; Chained matrix multiplication; Travelling Salesman Problem	5
4.2	Exploring Graphs: Traversing trees, Depth-first search, Breadth-first search; Backtracking; Branch-and-bound. An Introduction to NP Completeness: Introduction, Concept of P and NP; Polynomial Reduction; NP Completeness Proof of some problems.	5

List of Practicals

1. Write a program to perform operation count for a given pseudo code
2. Write a program to perform Bubble sort for any given list of numbers.
3. Write a program to perform Insertion sort for any given list of numbers.
4. Write a program to perform Quick Sort for the given list of integer values.
5. Write a program to find Maximum and Minimum of the given set of integer values.
6. Write a Program to perform Merge Sort on the given two lists of integer values.
7. Write a Program to perform Binary Search for a given set of integer values recursively and non recursively.
8. Write a program to find solution for knapsack problem using greedy method.
9. Write a program to find minimum cost spanning tree using Prim's Algorithm.
10. Write a program to find minimum cost spanning tree using Kruskal's Algorithm.
11. Write a program to perform Single source shortest path problem for a given graph.
12. Write a program to find solution for job sequencing with deadlines problem.

BCA -DC403 | Introduction to Cloud Technologies

Teaching Scheme			
Lecture hours per week	Tutorial hours per week	Practical Hours per week	Credits
2	1	2	4

Rationale

This course offers detailed concepts, applications, principles and implementation of cloud computing. It includes introduction, Cloud Computing Architecture, Cloud Virtualization, Cloud Programming Models, Cloud security and applications. It does not entirely focus on theoretical concepts but also strongly focuses on practical skill based learning.

Objective

The general objectives of this course are to provide theoretical as well as practical knowledge of cloud computing to make students capable of designing, implementing and managing the issues of cloud computing in their personal as well professional life.

Syllabus

Unit	Titles	Hours
Unit 1:	Introduction to Cloud Computing	8
1.1	Overview of Cloud Computing, Evolution of Cloud Computing,Characteristics of Cloud Computing, Types of cloud and its Cloud services, Benefits and challenges of cloud computing, Applications cloud computing, Cloud Storage, Cloud services requirements, cloud and dynamic infrastructure, Cloud adoption	
Unit 2 :	Cloud Computing Architecture	14
	Cloud reference model, Platform as service,Software as a service, Infrastructure as service,Cloud deployment models, Public clouds, Private clouds, Community cloud, Hybrid clouds, Cloud design and implementation using SOA,, security, trust and privacy	
Unit 3:	Cloud Virtualization technology	9

	Overview of Virtualization techniques, Types of Virtualizations,Implementation Levels of Virtualization Structure, virtualization benefits, server virtualization, hypervisor management software, virtual infrastructure requirements.Introduction to Security,Cloud Security challenges and Risks, Software-as-a-Service Security,Security Monitoring, Security Architecture Design, Data Security, Application Security, Virtual Machine Security, Identity Management and Access Control	
Unit 4:	Cloud platforms and applications	9
	Web services, AppEngine, Azures Platform,Aneka,Open challenges,Scientific applications,Business and Consumer applications	

Practical Works

1. The practical work consists of all features of cloud computing and field visit.
2. Visit the cloud service provider (cloud industries) nearby you and prepare a report based on organizational structure and technology implemented consulting with your subject teacher.

Text Books

- Dr. Kumar Saurabh, Cloud Computing
- Raj Kumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, Mastering Cloud Computing

Reference Books

- David S. Linthicum, Cloud Computing and SOA Convergence in your enterprise
- Barrie Sosinsky, Cloud Computing Bible
- Saurabh, K. (2011). Cloud Computing — Insights into New -Era Infrastructure, Wiley India.

BCA -DC404 | Cyber Security & Cyber Law

Teaching Scheme				
Lecture hours per week	Tutorial hours per week	Online Hours per week	Practical Hours per week	Credits
3	1	0	0	4

Objective

The exposure of the students to Cyber Security program at Graduate and Post Graduate level should lead to Learn the foundations of Cyber security and threat landscape and to equip students with the technical knowledge and skills needed to protect and defend against cyber threats.Also to develop skills in students that can help them plan, implement, and monitor cyber security mechanisms to ensure the protection of information technology assets.

Learning Outcome

Upon completion of the degree program, students will be able to understand the cyber security threat landscape and develop a deeper understanding and familiarity with various types of cyberattacks,cyber crimes, vulnerabilities and remedies thereto analyse and evaluate existing legal framework and laws on cyber security.

Syllabus

Unit	Title	Hours
Unit 1:	Definitions: Protection, Security, risk, threat, vulnerability, exploit, attack, confidentiality, integrity, availability, non-repudiation, authentication , authorization, codes, plain text, encryption, decryption, cipher text, key, ciphers, Symmetric and asymmetric cryptography, Public key , private key ,Crypt analysis,, Cyber forensics. Substitution cipher, Transposition cipher	10
Unit 2:	Risk analysis, process, key principles of conventional computer security, security policies, data protection, access control, internal vs external threat, security assurance, passwords, access control, computer forensics and incident response.	10
Unit : 3	CYBER ATTACKS (definitions and examples):	10

	Denial-of-service attacks, Man-in-the middle attack, Phishing, spoofing and spam attacks, Drive- by attack, Password attack, SQL injection attack, Cross-site scripting attack, Eavesdropping attack, Birthday attack, Malware attacks, Social Engineering attacks	
Unit 4 :	Firewalls, logging and intrusion detection systems, e-mail security, security issues in operating systems, ethics of hacking and cracking. Brief introduction of IT infrastructure in India, National agencies handling IT	10

Reference

- Merkow, M., & Breithaupt, J.(2005) Information Security Principles and Practices. 5th edition. Prentice Hall.
- Snyder, G.F. (2010). Network Security, Cengage Learning.
- Whitman, M. E. & Mattord, H. J. (2017) Principles of Information Security. 6th edition. Cengage Learning.

Additional Resources

- Basta, A., & Halton, W., (2010) Computer Security: Concepts, Issues and Implementation, Cengage Learning India.

- Charles P. Pfleeger, Shari Lawrence Pfleeger, Security in Computing,4 th Edition,
- Sushila Madan, Cyber Crimes and Laws, Scholar Tech Press (MKM Publishers Pvt. Ltd) Second Revised Edition, 2017

Online Resources

- https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/
- https://www.ibef.org/industry/infrastructure-sector-india.aspx
- https://www.ibm.com/in-en/topics/infrastructure
- https://business.mapsofindia.com/india-budget/infrastructure/it.html
- https://nasscom.in/knowledge-center/publications/it-infrastructure-services-digital-era
- https://digitalindia.gov.in/infrastructure
- https://techdifferences.com/difference-between-digital-signature-and-digital-certificate.html
- https://techdifferences.com/difference-between-digital-signature-and-electronic-signature.html

BCA -EC401 | Data Sciences Using Python

Teaching Scheme			
Lecture hours per week	Tutorial hours per week	Practical Hours per week	Credits
2	1	2	4

Objective

This course offers detailed concepts of data sciences with the history and preparation of data . It includes an introduction to Python and the implementation for the visualization of data.

Learning Outcome

The general objectives of this course are to provide theoretical as well as practical knowledge of techniques which are used in Data Sciences using python . After going through the course the student will be able to understand how and where which type of algorithm is applicable and how to get the maximum information for a data set under trial.

Syllabus

Unit	Titles	Hours
Unit 1 :	Overview of Python and Data Structures	
	Basics of Python including data types, variables, expressions, objects and functions. Python data structures including String, Array, List, Tuple, Set, Dictionary and operations.	
Unit 2 :	Data Science and Python	
	Discovering the match between data science and python: Defining the Sexiest Job of the 21st Century, Considering the emergence of data science, Outlining the core competencies of a data scientist, Linking data science, big data, and AI , Understanding the role of programming, Creating the Data Science Pipeline, Preparing the data, Performing exploratory data analysis, Learning from data, Visualizing, Obtaining insights and data products, Understanding Python's Role in Data Science, Considering the shifting profile of	

	data scientists, Working with a multipurpose, simple, and efficient language, Learning to Use Python Fast ,Loading data, Training a model, Viewing a result.	
	Introducing Python's Capabilities and Wonders: Why Python?, Grasping Python's Core Philosophy, Contributing to data science, Discovering present and future development goals, Working with Python, Getting a taste of the language, Understanding the need for indentation, Working at the command line or in the IDE, Performing Rapid Prototyping and Experimentation.	
Unit 3 :	Getting Your Hands Dirty With Data Understanding the tools:	
	Using the Jupyter Console, Interacting with screen text, Changing the window appearance, Getting Python help, Getting IPython help, Using magic functions, Discovering objects, Using Jupyter Notebook, Working with styles, Restarting the kernel, Restoring a checkpoint, Performing Multimedia and Graphic Integration, Embedding plots and other images, Loading examples from online sites, Obtaining online graphics and multimedia. Working with Real Data: Uploading, Streaming, and Sampling Data, Uploading small amounts of data into memory, Streaming large amounts of data into memory, Generating variations on image data, Sampling data in different ways, Accessing Data in Structured Flat-File Form ,Reading from a text file Reading CSV delimited format, Reading Excel and other Microsoft Office files, Sending Data in Unstructured File Form, Managing Data from Relational Databases, Interacting with Data from NoSQL Databases, Accessing Data from the Web.	
Unit 4:	Data Visualization	
	Visualizing Information: Starting with a Graph, Defining the plot, Drawing multiple lines and plots, Saving your work to disk, Setting the Axis, Ticks, Grids, Getting the axes, Formatting the axes, Adding grids, Defining the Line Appearance, Working with line style, Using colors, Adding markers, Using Labels, Annotations, and Legends, Adding labels, Annotating the chart, Creating a legend.	

Reference Books

- Python for data science for dummies 2nd Edition, John Paul Mueller, Luca Massaron, Wiley
- Programming through Python, M. T. Savaliya, R. K. Maurya, G. M. Magar, STAREDU Solutions
- Pandas for everyone :Python Data Analysis, Daniel Y. Chen, Pearson
- Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, Arno D.B. Meysman, et al., Minning
- Applied Data Science with Python and Jupyter: Use powerful industry-standard tools to unlock new, actionable insights from your data, , Packt
- Data Analytics, Anil Maheshwari , McGrawHill
- Data Science From Scratch: First Principles with Python, Joel Grus , SPD
- Star Data Science Specialist, STAR CERTIFICATION

BCA -EC402 | C# Programming

Teaching Scheme			
Lecture hours per week	Tutorial hours per week	Practical Hours per week	Credits
2	1	2	4

Pre- Requisites

Basic Programming Language

Rationale

Microsoft designed it as the official language of its .NET framework. Anything written in the .NET framework runs in Windows, which makes C# one of the official languages of Windows development. With the introduction of .NET Core, C# can now be used to create apps on macOS, Linux, and even Raspberry Pi. Microsoft is constantly adding new features to the language. Microsoft also provided support for seamless integration with other Microsoft technologies, such as Microsoft SQL server, cloud computing, Azure deployment and many more. Having these features makes the language more versatile, easy to learn and increases its usability.

Learning Outcome

- Understand basics, topologies and working mechanism of wired and wireless computer networks.
- Analyze the features and operations of protocols of OSI reference model & TCP/IP protocol suite.
- Design, calculate, and apply routing mechanisms for IPv4 & IPv6
- Identify the networking requirements for an organization and select & propose appropriate architecture and technologies.
- Work on Network addressing, design and implementation.

Syllabus

Unit	Title	Hours
Unit 1	The CLR and .NET Framework	08
1.1	Understand the motivation behind the .NET platform, Common Language Infrastructure (CLI). Know the role of the Common Type System (CTS), the Common Language Specification (CLS) and the Common Language Runtime (CLR)	4

1.2	Understand the assembly, metadata, namespace, type distinction, Contrast single-file and multi-file assemblies, Know the role of the Common Intermediate Language (CIL), Platform independent .NET (Mono / Portable .NET distributions).	4
Unit 2	C# Language Fundamentals	12
2.1	Language Fundamentals, Reference and value Types, primitive types the Nullable and enum types, Classes and objects, defining classes Creating objects, Using static members, Garbage Collector, Overloading Methods, Various Constructors. Encapsulating data, access modifiers, properties, indexers arrays and readonly fields.	7
2.2	Handling errors and throwing exceptions The Root object class. Inheritance and polymorphism specialization and generalization, Abstract classes, nesting of classes. Structures. String and DateTime classes.	5
Unit 3	Advanced Features	10
3.1	Event handling paradigm: Delegates and events. Anonymous delegates and lambda expression FUNC and Action delegates.	3
3.2	Generics Collections Interfaces, overriding interface implementation. Explicit interface implementation. Collection, IEnumerable, IEnumerator, IList, IComparer and their Generic equivalent. Working with generic List, Stack, Dictionary and Queue.	3
3.3	Programming Window Forms Applications: The notifies - subscribers paradigm for handling events. .NET framework for handling GUI events. Introduction to WPF and building an WPF application.	4
Unit 4	Introducing LINQ and XML	10
4.1	XML A quick introduction. LINQ and C#. Defining and executing a Query. Implicitly typed local variables. Anonymous Types, Extension Methods and Lambda Expressions. Putting LINQ to work. LINQ to SQL Fundamentals of ADO.NET Updating retrieving and deleting data using LINQ to SQL.	10

List of Practicals

1. To understand the Visual Studio 2010, its features, installation and overview.
2. Program to show the use of WriteLine(), Write(), ReadLine() and ReadKey() methods.
3. Program to perform various Arithmetic operations demonstrating use of switch statement.
4. Program to print sum of digits of a number, reverse the number and print multiplication table of the number (Use for and do-while loop).
5. Program to print Prime numbers in a user-defined range.
6. Program to print Fibonacci series of n numbers using recursion.
7. Program to print Factorial of a given number with and without using recursion.
8. Program to convert English Distance (Feets & Inches) to Metric Distance (Meters & Centimeters) and vice-versa using properties.
9. Program to declare a class EnglishDistance to store distance in Feets and Inches. Define necessary Constructors, Properties and Methods to find sum and difference of two distance objects.
10. Program to declare a class Date_Time to store date and time entered by user. Check whether user has entered valid date and time, and display message accordingly. Make use of necessary Constructors and properties.
11. Program to declare a class to store a Person's info such as ID, Name, Phone Number, DOB and Address. Use necessary Methods and Properties for taking values from user and setting up the values respectively. Override ToString() method to print the values to the screen.
12. Program to show the use of Abstract classes. Create Abstract Class Shapes for various shapes and create derived classes for shapes like Circle, Triangle, Rectangle, Square to find Area and Perimeter of the shape. Use Overriding of To String Method for printing the Data of Shape.
13. Program to implement array using Array class and implement BinarySearch() & Sort() methods.
14. Program to implement Method Overriding. Make use of 'virtual' and 'override' keywords.
15. Program to show the use of Abstract classes.
16. Programs on exception handling.
17. Program to create a Custom (User-Defined) Exception which is thrown when an invalid name is entered by the user. Also write code to catch and handle this Exception.
18. Programs demonstrating LINQ.
19. Program to find Odd & Even numbers in a given array using LINQ.
20. Program to create an XML document.
21. Program to create a Calculator using Windows Form.
22. Mini Project in C# (50% Weightage)

Learning Approach

1. Understand basics, topologies and working mechanism of wired and wireless computer networks.
2. In this course student will become familiar with a with C# language. This course will help to develop real life projects.

Reference Books

- Jesse Liberty and Donald Xie , “Programming C# 5.0”, O’REILLY.
- J.G.R. Sathiaseelan, N Sasikaladevi, “Programming with C# .net”, PHI, 2009.
- Paul J. Deitel, Harvey Deitel, “C# 2008 for Programmers”, Pearson, 3rd Ed., 2010.
- Joseph Albahari and Ben Albhari, “C# 5.0 in NUTSHELL”, O’REILLY.
- Herbert Schildt “C# 4.0: The Complete Reference”, McGraw-Hill
- Anders Hejlsberg Mads Torgersen Scott Wiltamuth Peter Golde, “The C# Programming Language Fourth Edition”, Microsoft Corporation Addison Wesley.

(b) Open source software and website address:

- https://onlinecourses.nptel.ac.in/noc21_cs18/preview

BCA-EC403 |ADMINISTRATION OF CLOUD
(Azure/ Google Cloud/ AWS)

Teaching Scheme			
Lecture hours per week	Tutorial hours per week	Practical Hours per week	Credits
2	1	2	4

Objectives

Cloud Computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.

Learning Outcome

The objective of the course is to learn about how the data can be distributed to the different data centres available to many users over the internet.

Syllabus

UNIT	Titles	Hours
Unit I :	Data Centre foot prints & Concepts ,Introduction To cloud , Virtualization concepts , Types of Virtualization & its benefits , Introduction to Various Virtualization OS Vmware , KVM etc HA/DR using Virtualization Moving VMs v. SAN backend concepts . Cloud Fundamentals Cloud Building Blocks Understanding Public & Private cloud environments (10 hrs)	10
UNIT-II	Cloud as IaaS a. Private Cloud Environment ,Basics of Private cloud infrastructure , QRM cloud demo , Public Cloud Environment i. Understanding & exploring Amazon Web services Managing and Creating Amazon EC2 instances , Managing and Creating Amazon EBS volumes , Tata Cloud details & demo	10
UNIT-III	Managing Hybrid Cloud environment 4. Setting up your own Cloud a. How to build private cloud using open source tools Understanding various cloud plug-ins Setting up your own cloud environment Auto provisioning Custom images Integrating tools like Nagios Integration of Public and Private cloud	10

Unit - IV	Future directions a. Cloud Domain and scope of work ,Cloud as PaaS, SaaS , Cloud Computing Programming Introduction Trends and market of cloud. Kubernetes (K8s) - Amazon Web Services (AWS) Database Migration Service - cloud extract, transform, load (ETL), Amazon SageMaker - platform to build, train and deploy machine learning models quickly; Cloud-based analytic databases - Amazon Redshift, Snowflake and Google BigQuery	10

Reference Books

Text Books:

1. Andreas M. Antonopoulos , “Mastering Bitcoin: Unlocking Digital Cryptocurrencies”, O’Reilly Media Inc, 2015
2. Melanie Swa “Blockchain”, First Edition, O’Reilly Jan 2015

Reference Books:

1. Cloud Computing: Fundamentals ByTimothy Chou's.
2. The Basics of Cloud Computing: Understanding the Fundamentals of Cloud Computing in Theory and Practice 1st Edition by Derrick Rountree (Author), Ileana Castrillo (Author)

List of Experiments

Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

1. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
2. Install Google App Engine. Create hello world app and other simple web applications using python/java.
3. Use GAE launcher to launch the web applications.
4. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
5. Find a procedure to transfer the files from one virtual machine to another virtual machine.
6. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
7. Install Hadoop single node cluster and run simple applications like word count.

BCA-EC404 | BlockChain Technologies

Teaching Scheme			
Lecture hours per week	Tutorial hours per week	Practical Hours per week	Credits
2	1	2	4

Objectives

This course offers detailed concepts, applications, principles and implementation of blockchain. It includes introduction,Architecture, usage and blockchain Programming Models, its applications. It does not entirely focus on theoretical concepts but also strongly focuses on practical skill based learning.

Learning Outcome

The general objectives of this course are to provide theoretical as well as practical knowledge of blockchain to make students capable of designing, implementing and managing the issues of in their personal as well professional life.

Syllabus

Unit	Title	Hours
Unit 1:	INTRODUCTION TO BLOCKCHAIN : What is Block chain? Basic ideas behind Blockchain, how it is changing the landscape of digitalization, Uses of Blockchain. Abstract Models for BLOCKCHAIN - GARAY model - RLA Model, what is Multichain? Objective of Multichain, Features of Multichain, Uses of Multichain, Process of mining in Multichain technology, Analyse Multichain platform, why it is better than other open platforms Blockchain Architecture and Design: Basic crypto primitives: Hash, Signature,) Hash chain to Blockchain, Basic consensus mechanisms. Practical Component Building and Deploying MultiChain private Blockchain	10
Unit 2 :	CONSENSUS & DAPPS : Requirements for the consensus protocols, Proof of Work (PoW), Scalability aspects of Blockchain consensus protocols Permissioned Blockchains: Design goals, Consensus protocols for Permissioned Blockchains (DAPPS) - Characteristics of Decentralized application, Setting up a Private Blockchain, Multiple configurable Blockchains using Multichain Deployment scenarios of Multichain, Centralized currency settlement, Bond issuance and peer-to-peer trading Consumerfacing rewards scheme in Decentralized Applications	10

	Practical Component 1. Deposit some Ether in your MetaMask accounts.	
	2. Create several accounts and make some transactions between these accounts	
Unit 3 :	Hyperledger Fabric (A): Decomposing the consensus process , Hyperledger fabric components, Chain code Design and Implementation Hyperledger Fabric (B): Beyond Chain code: fabric SDK and Front End (b) Hyperledger composer tool	10
	Practical Component 1. Creating a Business Network using Hyperledger 2. Creating a Business Network using Hyperledger – II	
Unit 4	USECASE MODEL – PRIVACY BLOCKCHAIN Use case 1: Blockchain in Financial Software and Systems (FSS): (i) Settlements, (ii) KYC, (iii) Capital markets, (iv) Insurance Use case 2: Blockchain in trade/supply chain: (i) Provenance of goods, visibility, trade/supply chain finance, invoice management discounting, etc.	10
	Practical Component Implementation of Use case – 1 & 2	

Reference Books

TEXT BOOKS

1. Andreas M. Antonopoulos , “Mastering Bitcoin: Unlocking Digital Cryptocurrencies”, O’Reilly Media Inc, 2015
2. Melanie Swa “Blockchain”, First Edition, O’Reilly Jan 2015

REFERENCE BOOKS/E-BOOKS

1. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric
2. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits - https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html
3. MOOC 1. https://www.udemy.com/course/build-blockchain/