



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.	Subject Code	Course Titles	Hours/week					Total Credits
			L	T	O	P	Total	
1	BCA-DC101	Mathematics - I (Calculus & Fourier Transformations)	3	1			4	4
2	BCA-DC102	Computer Organization and Architecture	2	1		2	5	4
3	BCA-DC103	Problem Solving & Programming in C	2	1		2	5	4
4	BCA-DC104	Information Technologies	2	1		2	5	4
5	BCA-DC105	Software Engineering	2	1		2	5	4
6	BCA-FW101	Face the World Skills-I	3				3	3
7	BCA-AE101	English Communication-I	2				2	2
Total			16	5		8	29	25

Semester II								
Sl No.	Subject Code	Course Titles	Hours/week					Total Credits
			L	T	O	P	Total	
1	BCA-DC201	Discrete Mathematics	3	1			4	4
2	BCA-DC202	Operating Systems	3	1			4	4
3	BCA-DC203	Free & Open Source Software (Linux Programming & Administration)	2	1		2	5	4
4	BCA-DC204	Data & File Structure using C	2	1		2	5	4
5	BCA-DC205	Object Oriented Programming using C++	2	1		2	5	4
6	BCA-AE201	Environmental Studies - I	2				2	2
7	BCA-AE202	English Communication-II	2				2	2
8	BCA-FW201	Face the World - II	3				3	3
Total			19	5		6	30	27

Semester III								
SI No	Subject Code	Course Titles	Hours/week					Total Credits
			L	T	O	P	Total	
1	BCA-DC301	Web Technologies & E- Commerce	2	1		2	5	4
2	BCA-DC302	Computer Graphics	2	1		2	5	4
3	BCA-DC303	Python Programming	2	1		2	5	4
4	BCA-DC304	Database Management Systems	2	1		2	5	4
5	BCA-DC305	Computer Networks	2	1		2	5	4
6	BCA-MC306	MOOC Course (Self Study)			2		2	1
7	FW-DG033	Face the World – III (Socio Emotional Learning)	2				2	2
8	FW-DG043	Face the World – III (Design Thinking)	1				1	1
9	AE-012	Environmental Studies-II	2				2	2
Total			15	5	2	10	32	26

Semester IV								
S I N O .	Subject Code	Course Titles	Hours/week					Total Credits
			L	T	O	P	Total	
1	BCA-DC401	Java Programming	2	1		2	5	4
2	BCA-DC402	Design and Analysis Algorithms	2	1		2	5	4
3	BCA-DC403	Introduction to Cloud Technologies	2	1		2	5	4
4	BCA-DC404	Cyber Security & Cyber Law	3	1			4	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		2	5	4
6	BCA-EC402	Elective - I C# Programming						
7	BCA-EC403	Elective – I						

		Administration of Cloud (Azure/Google Cloud/AWS)						
8	BCA-EC404	Elective - I BlockChain Technologies						
9	BCA-EC405	Elective - I C# Programming						
10	BCA-SM401	Seminar/ Research Paper Presentation				2	2	1
11	BCA-FW401	Personality Development	3				3	3
12	BCA-SE401	MOOC Course (Self Study)			2		2	1
Total			14	5	2	10	31	25

Semester V								
Sl No.	Subject Code	Course Titles	Hours/week					Total Credits
			L	T	O	P	Total	
1	BCA-DC501	Artificial Intelligence & Neural Networks	3	1			4	4
2	BCA-DC502	Machine Learning Using Python	2	1		2	5	4
3	BCA-DC503	Mobile Application Development	2	1		2	5	4
4	BCA-DC504	Software Project Management & SPM Tools	2	1		2	5	4
Students will be required to Select One of the Electives from Elective-II								
5	BCA-EC501	Elective - II Adv Java Programming	2	1		2	5	4

6	BCA-EC502	Elective - II Data Mining and Data Warehousing						
7	BCA-EC503	Elective - II ERP & ERP Tools						
8	BCA-EC504	Elective - II Programmi ng in R						
9	BCA-EC505	Elective - II IOT Applicat ion Development						
10	BCA-EC506	Elective - II Introduction to DevOps Technology						
11	BCA-PR501	Mini Project				4	4	2
12	BCA-FW501	Career Planning	3				3	3
13	BCA-SE501	MOOC Course (Self Study)			2		2	1
Total			14	5	2	12	33	26

Semester VI								
S I N O .	Subject Code	Course Titles	Hours/w eek					Total Credit s
			L	T	O	P	Tot al	
1	BCA-IT601	Research / Dissertation Project	1 4				14	14
2	BCA-SE601	MOOC Course (Self Study)			2		2	1
Total			14		2		16	15

#The syllabus for English Communication I and II, Face the World Skills and Environmental Studies are given separately.

SEMESTER I

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

Course Name: Mathematics - I	
Semester 1	
Course Code	BCA - DC101
Course Title	Mathematics -I (Calculus & Fourier Transforms)
Number of Credits	Total: (L: 3 T: 1)
Total Number of teaching hours	40 Hours
Pre-requisites	Basic Mathematical Concepts
Course Category	Core

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	Hou rs
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

Course Name: Computer Organization and Architecture	
Semester 1	
Course Code	BCA-DC102
Course Title	Computer Organization & Architecture
Number of Credits	Total: (L: 2 T: 1 P:2)
Total Number of teaching hours	40 Hours
Pre-requisites	Microprocessors and Digital Electronics
Course Category	Core

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 Microoperations, 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 Practical

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-Slave 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. William 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

Course Name: Problem Solving & Programming in C	
Semester 1	
Course Code	BCA-DC103
Course Title	Problem Solving & Programming in C
Number of Credits	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category	Core

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	Hou rs
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 Practical

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/>

Course Name: Information Technologies	
Semester 1	
Course Code	BCA-DC104
Course Title	Information Technologies
Number of Credits	Total: (L:2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Prerequisites	Basic Knowledge of Computer
Course Category	Core

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 Practical

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

Course Name: Software Engineering	
Semester 1	
Course Code	BCA-DC105
Course Title	Software Engineering
Number of Credits	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Prerequisites	Basic Knowledge of Computer & Software
Course Category	Core

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.

BCA - DC201 | Discrete Mathematics

Course Name: Discrete Mathematics	
Semester II	
Course Code	BCA-DC201
Course Title	Discrete Mathematics
Number of Credits	Total: (L: 3 T:1)
Total Number of teaching hours	40 Hours
Prerequisites	Basic Knowledge of Mathematics
Course Category	Core

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	Hou rs
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	4

	Databases.	
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 Congruences	2
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 : Swapn Kumar Sarkar
4. Discrete Mathematics : S.K. Chakraborty and B.K.Sarka

Course Name: Operating System	
Semester II	
Course Code	BCA-DC202
Course Title	Operating System
Number of Credits	Total: (L: 3 T:1 P: 0)
Total Number of teaching hours	40 Hours
Prerequisites	Basic Knowledge of Computer and programming
Course Category	Core

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.	

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)

Course Name: Free & Open Source Software (Linux Programming & Administration)	
Semester II	
Course Code	BCA-DC203
Course Title	Free & Open Source Software (Linux Programming & Administration)
Number of Credits	Total: (L: 2 T:1 P:2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category	Core

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	Hou rs
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 Practical

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

Course Name: Data & File Structure using C	
Semester II	
Course Code	BCA-DC204
Course Title	Data & File Structure using C
Number of Credits	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category	Core

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 Practical

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/>

Course Name: Object Oriented Programming using C++	
Semester II	
Course Code	BCA-DC205
Course Title	Object Oriented Programming using C++
Number of Credits	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	Introduction to Programming using 'C' Data Structure Concepts
Course Category	CORE

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	1 0
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	1 0
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	1 0
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 concept 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 Bhawe, "Object-Oriented Programming with C++", Pearson Education.
5. D . Parsons, "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

Course Name: Web Technologies & E-Commerce	
Semester 3	
Course Code:	BCA-DC301
Course Title:	Web Technologies & E- Commerce
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Prerequisites	NIL
Course Category:	Core

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

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 simple back-end database to support a website.
- Students should be able to recognize and evaluate website organizational structure and design elements.

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.	

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

Learning Approach

To understand HTML, PHP and Database connectivity with PHP.

References/suggested learning resources

Books

Textbooks:

- [T1] The complete reference HTML, by Thomas A powell, TMH publication.
- [T2] Mastering HTML 4.0 by Deborah S. Ray and Erich J. Ray. BPB Publication.
- [T3] Programming PHP. Rasmus Lerdorf, Kevin Tatroe. (O'Reilly, ISBN 1565926102).
- [T4] PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Websites Robin Nixon O'Reilly Media; 1 edition

Reference books:

- [R1] Core PHP Programming. Leon Atkinson (Prentice Hall, ISBN 0130463469).
- [R2] Beginning PHP5 and MySQL: From Novice to Professional, W. Jason Gilmore, 200 Apress, ISBN: 1-893115-51-8

Course Name: Computer Graphics	
Semester 3	
Course Code:	BCA- DC302
Course Title:	Computer Graphics
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Prerequisites	Basic Knowledge of Mathematics Concept & Programming Language Skill
Course Category:	Core

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

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.

nit	Title	Hou rs
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 Practical

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
4. 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, 2nd Edition, 2002.
2. Foley James D, "Computer Graphics: Principles and Practice", Addison-Wesley Professional, 3rd Edition, 2013.
3. Rogers and Adams, "Mathematical Elements for Computer Graphics", McGraw-Hill, 2nd Edition, 2002.

Reference books:

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

Course Name: Python Programming	
Semester 3	
Course Code	BCA-DC303
Course Title	Python Programming
Number of Credits	Total: (L: 2 T:1 P:2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category	Core

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

- 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.

Detailed Contents

Unit	Title	Hou rs
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 Practical

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

Course Name: Database Management System	
Semester 3	
Course Code	BCA-DC304
Course Title	Database Management System
Number of Credits	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	1. Elementary Maths (Sets, Relations) 2. Computer fundamentals related to memory organization such as primary memory, secondary memory, etc, 3. Knowledge of basic data structures. 4. Basic knowledge of data storage and file management system
Course Category	Core

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

- 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.

Detailed Contents

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 Practical

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:

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, —Database System ConceptsII, 5th Edition, Tata McGraw Hill, 2006
2. Elmasri and Navathe, —Fundamentals of Database SystemsII, 6th Ed., Pearson, 2013

3. C.J.Date, A.Kannan, S.Swamynathan, —An Introduction to Database Systemsll, 8th Edition, Pearson Education, 2006.
4. J. D. Ullman, —Principles of Database Systemsll, 2nd Ed, Galgotia Publications, 1999.
5. Vipin C. Desai, —An Introduction to Database Systemsll, West Publishing Co.

(b) Open source software and website address:

1. Oracle 10g
2. MySql

Course Name: Computer Networks	
Semester III	
Course Code	BCA-DC305
Course Title	Computer Networks
Number of Credits	Total: (L: 2 T:1 P:2)
Total Number of teaching hours	40 Hours
Pre-requisites	Basic Fundamentals of Computers
Course Category	Core

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.

Detailed Contents

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	5

	Control Protocol, Performance Issues, QOS, Transport Layer Security, IP Security, IPSec	
4 2	<p>Application Layer: DNS, Electronic Mail, WWW, Streaming Protocols, DHCP & DHCPv6, SMTP, Telnet, SSH, POP, POP3, IMAP</p> <p>Network Security: Need for Network Security, Cryptography and Compression Techniques, Firewall, Various types of Cyber Threats</p>	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 Graw 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>

Course Name: JAVA PROGRAMMING	
Semester IV	
Course Code	BCA-DC401
Course Title	JAVA PROGRAMMING
Number of Credits	Total: (L: 2 T: 1 P:2)
Total Number of teaching hours	40 Hours
Pre-requisites	Object Oriented Concepts
Course Category	Core

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.

Detailed Contents

Unit	Title	Hours
Unit 1	Basics of Java programming	12
1.1	Java Programming: History and Features of Java, C++ vs. Java, JDK, JRE, and JVM (Java Virtual Machine), Java Tokens: Data types, Literals and Variables, Operators, Control Structures, Arrays: 1D, 2D and Multidimensional Arrays, Array Class, Concept of Dynamic Arrays Classes: Fundamentals, objects, Creating a class, methods, Access modifiers, Constructors, this keyword, static (variable, method, block), final keyword - final variables, Wrapper Classes, Garbage Collection	6
1.2	Polymorphism: Method overloading, constructor overloading.	6

	<p>Inheritance: Super class, subclass, Types of Inheritance: Single-level Inheritance, Multi-level Inheritance, Hierarchical Inheritance, Multiple Inheritance using interface, Hybrid Inheritance, super keyword, method overriding, use of final methods & classes), abstract class</p> <p>Interfaces: Creating and implementing an interface, difference between abstract class & interface</p> <p>Packages: Concept, Importing a package and creation of the custom package.</p>	
UNIT 2	Exception Handling and Multithreading	8
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 class, StringBuffer class and StringBuilder class and comparison among them.	2
UNIT 3	Streams, GUI Programming & Event Handling	12
3.1	Streams & Elementary concepts of Input/Output: Byte and Character streams, reading & writing using byte & character streams, System.in and System.out, print and println, reading from a file and writing in a file.	3
3.2	<p>Applets: Introduction and life cycle.</p> <p>AWT controls: Button, Label, TextField, TextArea, Choice lists, list, scrollbars, check boxes, Layout managers</p> <p>Swings: Introduction, Components (JLabel, ImageIcon, JButton, JToggleButton, JCheckBox, JRadioButton, JTextField, JTextArea, JPasswordField, JList, JComboBox, JScrollPane), implementation and comparison with AWT.</p>	6
3.2	Elementary concepts of Event Handling: Delegation Event Model, Event classes, interfaces and listeners, Adapter classes, Inner classes.	3
UNIT 4	Networking, Database and Servlets	8
4.1	<p>Networking Basics: Socket (datagram and TCP/IP based client and server socket), factory methods, InetAddress</p> <p>JDBC: JDBC Architecture, JDBC Drivers, Connection, Statement, PreparedStatement, ResultSet, Connecting to the Database.</p>	5
4.2	Introduction to Java Servlets: Life cycle, Interfaces and classes in javax, servlet package (only description), Creating a simple servlet.	3

List of Practical:

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 learning resources:

Books

TEXT BOOKS:

- [T1] Patrick Naughton and Herbert Schildt, "Java-2 The Complete Reference", TMH.
- [T2] Y. Daniel Liang, "Introduction to Java Programming, Comprehensive Version, 7/e"
Pearson

REFERENCE BOOKS: -

- [R1] Krishnamoorthy R, Prabhu S , "Internet and Java Programming", New Age Intl.
- [R2] David Flanagan, Jim Farley, William Crawford and Kris Magnusson, "Java Enterprise in a Nutshell", O'Reilly.

Course Name: C# Programming	
Semester IV	
Course Code:	BCA-EC402
Course Title:	Elective - I C# Programming
Number of Credits:	Total: (L: 3 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	Basic Programming Language
Course Category:	Elective

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 the concept of DOT NET Framework.
- Understand the management of Data Structure through DOT NET Framework.
- Understand what is Garbage Collection.
- C# Programming language concepts.
- Object Oriented Programming through C#. Use of Inheritance, Multiple & Multi Level Inheritance & Interface.
- Concepts of Abstract Classes & Abstract Functions, properties, Overloading, Over riding, Exception Handling, Database handling, File handling.
- Formatted Input Output handling through C#.
- Windows Application development & Web application development through C#.

Detailed Contents

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. In this course student will become familiar with a with C# language. This course will help to develop real life projects and Windows Application as well as Web applications.
2. Initially, Basic concepts of DOT NET framework may be taken up, followed by C# Language concepts.

References/suggested learning resources:

(a) Books

1. Jesse Liberty and Donald Xie , “Programming C# 5.0”, O'REILLY.
2. J.G.R. Sathiaselvan, N Sasikaladevi, “Programming with C# .net”, PHI, 2009.
3. Paul J. Deitel, Harvey Deitel, “C# 2008 for Programmers”, Pearson, 3rd Ed., 2010.
4. Joseph Albahari and Ben Albahari, “C# 5.0 in NUTSHELL”, O'REILLY.
5. Herbert Schildt “C# 4.0: The Complete Reference”, McGraw-Hill
6. 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:

1. <https://www.csharp411.com/best-c-web-sites/>
2. <https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps>
3. <https://www.mono-project.com/>



Course Name: Artificial Intelligence & Neural Networks	
Semester V	
Course Code:	BCA-DC501
Course Title:	Artificial Intelligence & Neural Networks
Number of Credits:	Total: (L: 3 T:1)
Total Number of teaching hours	40 Hours
Pre-requisites	
Course Category:	Core

Rationale:

Artificial Intelligence is utilized heavily in computizing cognitive functions such as speech and Vision. Often these functions are achieved through the use of Neural networks. In this module, we will study very popular Neural Network architectures for achieving various cognitive functions such as Object recognition, natural language processing besides explores reinforcement learning. We will study and practice various use cases ranging from text generation, object tagging to fraud detection and learning games.

Learning Outcome:

- Students will learn the basic concepts and techniques of Artificial Intelligence. They should be able to develop AI algorithms for solving practical problems.

Detailed Contents

Unit	Title	Hours
Unit 1	Foundations for AI	10
1	Defining AI techniques, Comparison - AI, ML and Deep Learning, Artificial Intelligence and its applications, Artificial Intelligence Techniques, Level of models, criteria of success, Intelligent Agents, Nature of Agents, Learning Agents. AI Techniques, advantages, and limitations of AI, Impact and Examples of AI, Application domains of AI.	
Unit 2	Problem solving techniques & Logic	10
2	State space search, control strategies, heuristic search, problem characteristics, production system characteristics., Generate and test, Hill climbing, best first search, A* search, Constraint satisfaction problem, Mean-end analysis, Min-Max Search, Alpha-Beta Pruning. Propositional logic, predicate logic, Resolution, Resolution in propositional logic and predicate logic, Clause form, unification algorithm	
Unit 3	Convolution Neural Networks	10

3	Image classification, Text classification, Image classification and hyper-parameter tuning, Emerging NN architectures: ResNet, AlexNet – Applications	
Unit 4	Recurrent Neural Networks	10
4	Recurrent Neural Networks, Building recurrent NN, Long Short-Term Memory, Time Series Forecasting, Bidirectional RNNs, Encoder-decoder sequence to sequence architectures - BPTT for training RNN, Computer Vision - Speech Recognition - Natural language Processing, Case studies in classification, Regression and deep networks.	10

Learning Approach:

- To understand Principles of Data Warehousing and Data Mining.
- To understand Architecture of Data Warehouse.
- To understand Various Data preprocessing Methods.
- To understand Data Mining techniques.

References/suggested learning resources:

TextBook and Learning Resource:

- A Classical Approach to Artificial Intelligence, M.C. Trivedi, Khanna Book Publishing, 2019.
- Artificial Intelligence: A modern approach by Stuart Russel, Pearson Education, 2010.
- Artificial Intelligence by Rich and Knight, The McGraw Hill, 2017.
- Artificial Intelligence: A new synthesis by Nils and Nilson, Elsevier, 1997.
- Artificial Intelligence by Luger, Pearson Education, 2002.
- Artificial Intelligence by Padhy, Oxford Press, 2005.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, “Deep Learning”, First Edition, MIT Press, 2016.
- Nikhil Buduma and Nicholas Lacascio, “Fundamentals of Deep Learning”, First Edition, O.Reilly, 2017
- Russel,S., and Norvig,P., (2015), Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall

Reference Books:

- E. RICH, K. KNIGHT, S. B. NAIR (2017), Artificial Intelligence, McGraw Hill Education, 3rd Edition.
- R.S. SUTTON, A.G. BARTO (2015), Reinforcement Learning: An Introduction, The MIT Press, 2nd Edition.
- Josh Patterson, Adam Gibson "Deep Learning: A Practitioner's Approach", O'Reilly Media, 2017
- Laura Graesser, Wah Loon Keng "Foundations of Deep Reinforcement Learning: Theory and Practice in Python" Addison-Wesley Professional -2020
- Jon Krohn, Grant Beyleveld, Aglaé Bassens "Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence", 1st edition Addison-Wesley Professional 2019

(b) Open source software and website address:

- <https://www.edx.org/course/artificial-intelligence-ai>
- <https://www.udemy.com/course/artificial-intelligence-az/>
- Department of Computer Science, University of California, Berkeley, <http://www.youtube.com/playlist?list=PLD52D2B739E4D1C5F>

- An Introduction to Artificial Intelligence By Prof. Mausam, IIT Delhi
- Artificial Intelligence, Prof. Sudeshna Sarkar, IIT Kharagpur

Tutorial List:

- Numerical type questions on CNN-
 - a. Parameters tuning
 - b. Convolution function
 - c. Different types of filters
- Fuzzy Logic and Neural Networks

Course Name:	
Semester V	
Course Code:	BCA-DC502
Course Title:	Machine Learning Using Python
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	Programming in Python
Course Category:	Core

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:

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

- i. Use python libraries for performing machine learning on datasets.
- ii. Understand and apply the machine learning workflow on datasets.
- iii. Implement data preprocessing on datasets.
- iv. Implement machine learning algorithms.
- v. Determine the performance of machine learning algorithms and evaluate the models.

Detailed Contents

Unit	Title	Hours
Unit I	Essential Python libraries for Machine Learning and Introduction to Machine Learning	8
	<p>Essential Python libraries for Machine Learning:</p> <p>Numpy: Creation of Numpy arrays, array indexing, slicing, reshaping, array math, assignment, Manipulating Tabular Data using Pandas: Pandas Series and DataFrame, Data Visualization using matplotlib, seaborn: Basics, line charts, bar charts, pie charts, scatter plots, subplots, Scikit-learn: Introduction, Getting the datasets, Features and Applications</p> <p>Introduction to Machine Learning:</p> <p>Basics, Significance, Problems ML can solve, Classification of Machine Learning Techniques</p>	
Unit 2	Machine Learning Workflow, Data Preprocessing and Supervised Learning Techniques	14
	<p>Machine Learning Workflow and Data Preprocessing: Machine Learning Lifecycle, Types of Data in datasets, Dataset Loading, Data Preprocessing: Data Outlier analysis, treating for missing values, encoding categorical data, splitting the dataset into training and test dataset, feature scaling, Feature Selection Techniques: Filter, Wrapper and Embedded methods.</p> <p>Supervised Learning Techniques: Introduction, Classification of supervised learning techniques, Linear Regression, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines, Naive Bayes Algorithm, Decision Tree, Model evaluation - accuracy, recall, precision, F1 score and Confusion matrices. Implementation of the learning techniques and determination of performance.</p>	
Unit 3	Unsupervised Machine Learning Techniques	8
	<p>Unsupervised Learning Techniques: Approaches: Clustering, Association, Dimensionality reduction, K-means, Hierarchical clustering, Principal Component Analysis, Frequent Itemset Mining, Apriori algorithm, Implementation of the learning techniques and determination of performance.</p>	
Unit 4	Ensemble Methods, Reinforcement Learning & Case studies on Machine Learning	8

	<p>Ensemble Methods & Reinforcement Learning: Bagging, Boosting, Random Forest, Reinforcement Learning: RL Framework and TD Learning</p> <p>Case Studies on Machine Learning: Analyzation on datasets using ML workflow.</p>	
--	--	--

List Of Practical:

Practical based on:

- i. Python libraries for machine learning
- ii. Dataset loading
- iii. Data preprocessing
- iv. Machine learning algorithms
- v. Dataset analysis using ML techniques.
- vi. Performance prediction

Learning Approach:

To learn & apply the fundamentals of python libraries for machine learning, machine learning workflow, data preprocessing, data analysis using ML algorithms.

References/suggested learning resources:

1. Python Machine Learning, Wei-Meng Lee, Wiley (2019)
2. Introduction to Machine Learning with Python, Andreas C. Müller, Sarah Guidoy, O'Reilly Media, Inc. (2016)
3. Machine Learning for Absolute Beginners, Oliver Theobald, Scatterplot Press (2021)
4. Machine Learning with Python Cookbook, Chris Albon, O'Reilly Media, Inc. (2018)
5. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Third Edition, Aurélien Géron, Shroff/O'Reilly (2022)
6. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies by John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy, MIT Press (2015)

Course Name: Mobile Application Development	
Semester V	
Course Code:	BCA-DC503
Course Title:	Mobile Applications Development
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category:	Core

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:

- Student knows mobile devices and mobile platforms
- Student has knowledge concerning mobile operating systems and their architecture
- Student is familiar with wireless communications standards and data transmission standards
- Student knows how to prepare a mobile application for distribution In terms of skills.
- Student can recognize and setup a mobile device and application runtime environment
- Student is able to setup programming tools for a mobile application developer (for selected modern mobile platforms)
- Student can select appropriate data transmission standards In terms of social competence.
- Student understands the need for continuous improvement of his/her skills due to the rapidly changing environment of mobile devices.

Detailed Contents

Unit	Title	Hours
Unit 1	Introduction to Web Application Development	10
1.1	Overview of Internet and web, <u>HTML Tags</u> , Forms & Frames , Introduction to <u>Java Script and Cascading Style Sheets</u> , <u>DHTML</u> .	

2m

1.2	Introduction to <u>PHP</u> , <u>MySQL/MariaDB</u> , Introduction to Web Server, Installation & Configuration of <u>WAMP/LAMP/XAMPP</u> , IIS. Developing Web Applications in PHP with Databases, Introduction to <u>BootStrap</u> .	
Unit 2	Android Systems	10
2.1	Android Systems: Introduction to Mobile devices and applications, <u>Open Handset Alliance (OHA)</u> , Overview of <u>Android OS and architecture</u> , installing android studio, understanding the android build system.	
2.2	Introduction to <u>Android Application Components</u> : Activity, service, content provider and broadcast received, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, <u>Using Intent Filter</u> , <u>Permissions</u> . <u>Activities and Intents</u> : Declaring and starting a new activity, switching between activities, passing data and returning value from activity, state of activity, storing persistent activity data, understanding <u>activity life cycle</u> , <u>Types of intents</u> , intent filter, context, data sharing using intent.	
Unit 3	Android Web Application Development	10
3.1	Android User Interface : Basic android UI, android layouts, view attributes, Objects & Controls and uses UI events and event listeners, <u>animations</u> , notifications, progress dialog, <u>Action bar</u> , toolbar, menus and pop ups, <u>Floating action button</u> , Tab based UI,	
3.2	Fragment, <u>Types of Fragment</u> , <u>Fragment Lifecycle</u> , communication between fragment and activity, add, replaced remove fragment, Listview, material design, recyclerview.	
Unit 4	Android Storage and API	10
4.1	<u>Android Storage</u> : Using Android Data and Storage APIs, Managing data using <u>SQLite</u> , Sharing Data between Applications with Content Providers <u>Android APIs</u> : Multimedia, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, android location based services	

List of Practical:

1. Installation of Android studio.
2. Development Of Hello World Application
3. Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button
4. Create a screen that has input boxes for User Name, Password, Address, Gender(radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout)
5. Design an android application to create page using Intent and one Button and pass the Values from one Activity to second Activity

6. Design an android application Send SMS using Intent
7. Create an android application using Fragments
8. Design an android application Using Radio buttons
9. Design an android application for the menu.
10. Create a user registration application that stores the user details in a database table.

Learning Approach:

1. Design scripts to meet given interface requirements.
2. Use variables, properties and other code elements appropriately to implement the code design.
3. Devise, carry out and evaluate functional test strategies of mobile design.
4. Implement and evaluate techniques for the installation of mobile applications and delivery via various channels.

References/suggested learning resources:

(a) Books

1. Abraham Silberschatz and Peter Baer Galvin, "Operating System Concepts", Addison-Wesley
2. Milan Milankovic, "Operating Systems, Concepts and Design", Tata McGraw-Hill.
3. Harvey M Deital, "Operating Systems", Addison Wesley
3. Richard Peterson, "Linux: The Complete Reference", Osborne Tata McGraw-Hill.
4. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education
5. D M Dhamdhare, "Operating Systems : A Concept based Approach", McGraw Hill.
6. Charles Crowley, "Operating Systems: A Design-Oriented Approach", Tata McGraw Hill Education".
7. Stuart E. Madnick & John J. Donovan. Operating Systems. McGraw Hill.

Course Name: Software Project Management & SPM Tools	
Semester V	
Course Code:	BCA-DC504
Course Title:	Software Project Management & SPM Tools
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	32 Hours
Pre-requisites	Basic knowledge of Software Engineering
Course Category:	Core

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 the concept of Software Project Management
- To provide fundamental knowledge of software project management and corresponding software tools.

Detailed Contents

Unit	Title	Hours
Unit 1	Software Project Management Concepts	08
1	Introduction, Project and Software project, Software project vs other project, Importance and Problems in software project management, Process of SPM. Characteristics of good project manager, Successful Software Project Manager, Overview of Software Project Planning.	
Unit 2	Software Project Scheduling & Cost Estimation	10
2	Objectives of activity planning, Work breakdown structure, Network planning model: Critical path method (CPM), Program evaluation and review technique (PERT), Precedence diagramming method (PDM), Shortening project duration, Identifying critical activities. Software Effort Estimation: Basis of software Estimating, Software effort estimation techniques expert Judgment, Estimating by analogy. Bottoms-up estimating, Top-down approach and parametric models. <i>NETWORK</i>	
Unit 3	Project Evaluation & Risk Management	10

3	Project Evaluation: Strategic Assessment, Technical Assessment, cost-benefit analysis, Risk Evaluation. Selection of Appropriate Report, Project approach: Choosing technologies, choice of process models, structured methods. Risk Identification, Planning, Evaluation and Management, Categories of Risk, Framework for dealing with risk, evaluating Risks to the schedule.	
Unit 4	Software Quality & Configuration Management	12
4	TQM, Six Sigma, SEI CMM, McCall Models for Software Quality, Defining and importance of software quality, Concept, Requirement and Elements of SCM, Baseline, SCM Repository, Versioning and version control, SCM Process, Change Control Process. Configuration Audit and Status Reporting. Study of Software Project Management tools.	

List Of Practical:

Laboratory works should be done covering all the topics listed above and a small work should be carried out using the concept learnt in each unit in the group. Work should be assigned on an individual basis. Students may choose project Management tools like (MS Project, OpenProj, dot Project, Trello, Asana, ClickUp etc.).

Learning Approach:

This course provides the comprehensive knowledge about Software Project Management, which encompasses with Software Project Planning, Scheduling, Cost Estimation, Risk management, Quality management and Configuration management

References/suggested learning resources:

1. Cotterell, B. H. (2018). Software Project Management. McGraw-Hill
2. Dutt, S. C. (n.d.). Software Project Management. Pearson Education India.
3. A.S. Kelkar (Node). Software Project Management. PHI Learning.

Course Name: Advance Java Programming	
Semester V	
Course Code:	BCA-EC501
Course Title:	Advance Java Programming
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category:	Elective

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:

- Student knows java applications in web development
- Dynamic web pages using Servlets
- Session management
- Designing custom tags using JSP
- Data access, searches and queries using HQL
- Web Application development using Spring framework
- Handling crosscutting concerns using Spring AOP

Detailed Contents

Unit	Title	Hours
Unit 1	Introduction to J2EE	10

1.1	Distributed Web Application Multi-Tier Architecture, MVC architecture, HTTP Basics, Introduction to servlets and its life cycle, problems with CGI/Perl interface, generic and http servlet, servlet configuration,	
1.2	Various session tracking techniques, servlet context, servlet configuration, servlet listeners, filters and annotations	
Unit 2	JSP Basics and Architecture & Struts Framework:	10
2.1	JSP directives, Scripting elements, standard actions, implicit objects, JSP design strategies. Using JSP Beans, Expression languages, JSP custom tags	
2.2	Introduction of Struts and its architecture, advantages, and application of Struts.	
Unit 3	Hibernate Framework:	10
3.1	Comparison between JDBC and Hibernate, Principles of Object Relational Mapping, Hibernate configuration, mapping and annotations,	
3.2	Session management, searches and queries using HQL, Transaction integration and demarcation.	
Unit 4	Spring Framework:	10
4.1	Introduction of Spring Framework: Configuration of Spring environment, Spring Architecture, Dependency Injection,	
4.2	Inversion of Control (IoC), Spring Boot, Using Hibernate with Spring, Spring AOP and Spring Web.	

List of Practical:

As per the Subject teacher assigned as per theory curriculum.

Learning Approach:

1. Model Java EE architectural components, Servlet creation and session management for web applications.
2. Inspect the fundamentals of Java Server Pages (JSP) and Struts for web-based applications.
3. Appraise Hibernate Framework of JEE and apply constructs of Object Relational Mapping.
4. Elaborate principles of Dependency Injection and its application in JEE.
5. Design applications based on Spring Boot and Spring AOP

1. (a) Books

TEXT BOOKS:

TB1. John Hunt and Chris Loftus, "Guide to J2EE: Enterprise Java", Springer Verlag Publications, 1st Edition, 2003.

TB2. Govind Sesadri, "Enterprise Java Computing: Application and Architectures", Cambridge University Publications, 2nd Edition, 1999.

TB3. Jeff Linwood and Dave Minter, "Beginning Hibernate", Apress Publishing Co., 2nd Edition, 2010.

TB4. Rod Johnson, "Professional Java Development with the Spring Framework", John Wiley & Sons, 2nd Edition, 2005.

REFERENCE BOOKS:

RB1. Ted Neward, "Effective Enterprise Java", Pearson Education, 2nd Edition, 2004.

RB2. Jim Farley and William Crawford, "Java Enterprise in a Nutshell", O'Reilly and Associates, 3rd Edition, 2005.

RB3. James Holmes and Herbert Schildt, "The Complete Reference- Struts", Tata McGraw Hill, 2nd Edition, 2007.

Course Name: Data warehouse and Data mining	
Semester V	
Course Code:	BCA-EC502
Course Title:	Data Warehouse and Data mining
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	32 Hours
Pre-requisites	Basic knowledge of DBMS
Course Category:	Core

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 the concept of Data Mining

Detailed Contents

Unit	Title	Hours
Unit 1	Data mining and Data preprocessing	10
1	Data mining: Introduction, Data mining – on what kind of data, data mining functionalities – what kind of patterns to be mined, Classification of data mining systems, data mining task primitives, integration of a data mining systems with a database or data warehouse systems, major issues in data mining. Data preprocessing: Descriptive data summarization, data cleaning, data integration and transformation, data reduction, data discretization and concept hierarchy generation	
Unit 2	Data warehouse and OLAP technology	10
2	Data warehouse and OLAP technology: What is data warehouse, A multidimensional data model, data warehouse architecture, data warehouse implementation, data warehouse usage, OLAP, OLAM Mining frequent patterns, association and correlation, efficient and scalable frequent itemset mining methods, From association mining to correlation analysis	
Unit 3	Classification and Cluster Analysis	10

3	Classification and prediction: Introduction, issues, classification by decision tree induction, rule based classification, classification by back propagation, lazy learners, other classification methods, Prediction: accuracy and error measures, evaluating the accuracy of a classifier or predictor. Cluster Analysis: Types of data in cluster analysis, a categorization of major clustering methods, partitioning methods.	
Unit 4	Mining complex types & Applications and trends in data mining	10
4	Mining complex types of data: Multidimensional analysis and descriptive mining of complex data objects, mining spatial database, multimedia database, mining world wide web. Applications and trends in data mining: Data mining applications, data mining system products and research prototypes, social impact of data mining, trends in data mining.	10

List Of Practicals :

- Practical based on demonstrate various OLAP operation on multi-dimensional data
- Practical Based on Apply data mining algorithms on real time data
- Practical Based on Supervised & Un Supervised Learning on Real Time Problems.
- Practical Based on Assess most appropriate data mining algorithm for given situation

Learning Approach:

- To understand Principles of Data Warehousing and Data Mining.
- To understand Architecture of Data Warehouse.
- To understand Various Data Preprocessing Methods.
- To understand Data Mining techniques.

References/suggested learning resources:

Text Book and Learning Resource:

- Data Warehousing (Fundamentals for IT Professionals) : Paulraj Ponniah
- Data Mining and WareHousing : M.SudheepElayidom
- Data Mining : Vikram Pudi, P. Radha Krishna
- Data Warehousing : Reema Thareja

Reference Books :

- Margaret Dunham, “ Data Mining: Introductory and Advanced Topics, 1/e”, Pearson
- G. K. Gupta, “Introduction to Data Mining with Case Studies”, PHI, 2006.
- W. H. Inmon, “Building the Operational Data Store”,2nd Ed., John Wiley, 1999
- B. M. Shawkat Ali, Saleh A. Wasimi, “Data Mining Methods and Techniques”, Cengage Learning, 2010

(b) Open source software and website address:

Course Name: ERP & ERP Tools	
Semester V	
Course Code:	BCA-EC503
Course Title:	ERP & ERP Tools
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	Basic knowledge of Business & Information System
Course Category:	Core

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:

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

- Understand the ERP Fundamentals.
- Understand ERP Functional Modules .
- Understand ERP Solutions..
- Understand ERP Implementation .

Detailed Contents

Unit	Title	Hours
Unit 1	Introduction to ERP	08
	Introduction: Overview of enterprise systems ñ Evolution - Risks and benefits - Fundamental technology - Issues to be considered in planning design and implementation of cross functional integrated ERP systems .	
Unit 2	ERP Solutions and Functional Modules	10
	ERP Solutions and Functional Modules: Overview of ERP software solutions- Small, medium and large enterprise vendor solutions, BPR and best business practices - Business process Management, Functional modules.	
Unit 3	ERP Implementation	10

	ERP Implementation: Planning Evaluation and selection of ERP systems - Implementation life cycle - ERP implementation, Methodology and Framework- Training ñ Data Migration - People Organization in implementation-Consultants, Vendors and Employees.	
Unit 4	Post Implementation & Emerging Trends in ERP:	12
	Post Implementation: Maintenance of ERP- Organizational and Industrial impact; Success and Failure factors of ERP Implementation. Emerging Trends on ERP: Extended ERP systems and ERP add-ons -CRM, SCM, Business analytics - Future trends in ERP systems-web enabled, Wireless technologies, cloud computing.	

List Of Practical:

- Introduction to ERP System and Review on different ERP packages.
- Study on Odoo ERP package.
- Identification and brief introduction of different Modules in Odoo ERP package.
- Detail study of odooerp modules.
- Identify different processes of the odooerp modules.
- Study on odooerp business functions

Learning Approach:

- To know the basics of ERP To understand the key implementation issues of ERP
- To know the business modules of ERP
- To be aware of some popular products in the area of ERP
- To appreciate the current and future trends in ERP

References/suggested learning resources:

1. Alexis Leon, ERP demystified, second Edition Tata McGraw-Hill,2008.
2. Sinha P. Magal and Jeffery Word, Essentials of Business Process and Information System, Wiley India,2012
3. Jagan Nathan Vaman, ERP in Practice, Tata McGraw-Hill,2008
4. Alexis Leon, Enterprise Resource Planning, second edition, Tata McGraw-Hill,2008.
5. Mahadeo Jaiswal and Ganesh Vanapalli, ERP Macmillan India,2009
6. Vinod Kumar Grag and N.K. Venkitakrishnan, ERP- Concepts and Practice, PHI,2006.

Course Name: Programming in R	
Semester V	
Course Code:	BCA-EC504
Course Title:	Programming in R
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	
Course Category:	Core

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:

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

- Understand the R Fundamentals.
- Write R scripts.
- Apply R data structure, functions and OOPS concepts.
- Use R to handle the data, manipulate and visualize.
- Use R for performing statistics on data and design GUI.

Detailed Contents

Unit	Title	Hours
Unit 1	R Fundamentals	10
	<p>Fundamentals: Features of R, Significance and advantages over other programming language, Literals, Variables, Operators in R, Data types, Operators: Arithmetic, Assignment, Logical, Relational, Miscellaneous, Keywords</p> <p>Decision Making and Looping: if statement, if...else statement, switch statement, if...else Ladder, ifelse() function, for loop, while Loop, repeat Loop, break Statement, next Statement, nested loops.</p>	
Unit 2	R Data Structures, Functions and Object-Oriented Programming	10
	<p>R Data Structures: Scalars, Vectors, Matrices, List, Dataframes, Matrices, Arrays, Factors, Creation, Access, and Manipulation of the data structures</p> <p>R Functions: Types, user defined function, variable scope.</p>	

	R Objects & Class Inheritance: Object, Class: S3, S4, Reference Classes, Inheritance Types.	
Unit 3	Data Handling and Visualization in R	14
	<p>Importing & Manipulating Data: Reading Tabular Data files, Reading CSV files, Importing data from excel, Analyzing CSV and Excel files, Accessing database, Saving in R data, Loading R data objects, Writing to files, Selecting rows/observations, Selecting columns/fields, Merging data, Converting variable types, Data sorting, Data aggregation</p> <p>Data Visualization: Bar graph, Histogram chart, pie graph, box chart, strip chart, Plot Function, Sub Plot Saving plot, Color in Graphs</p>	
Unit 4	Statistics for Data Science & Designing GUI	6
	<p>Statistics for data science: Mean, Median, Mode, Standard Deviation, Variance, Data Analysis</p> <p>Designing GUI: Building interactive application and connecting it with database.</p>	

List Of Practical:

Practical based on:

- i. R Fundamentals – operators, data types.
- ii. Decision Making and Looping.
- iii. R data structures, functions, OOPS concepts.
- iv. Reading Tabular, CSV and Excel files, accessing database, manipulating and visualizing data.
- v. Statistics and GUI programming using R.

Learning Approach:

To learn & apply the fundamentals of R programming, data structures, OOPS concepts, handling data, statistics, analyzing, visualizing and performing GUI programming.

References/suggested learning resources:

1. Andrie de Vries, Joris Meys, R for Dummies A Wiley Brand, 2nd Edition, John Wiley and Sons, Inc, 2015, ISBN-13 : 978-1119055808
2. Garrett Grolmund, Hands on Programming With R: Write Your Own Functions and Simulations, Shroff/O'Reilly, 2014
3. Sandip Rakshit, R Programming for Beginners, McGraw Hill Education (2017)
4. JD Long et al., R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics, O'Reilly Media (2019)
5. Richard Cotton, Learning R: A Step-by-Step Function Guide to Data Analysis, Shroff/O'Reilly; First Edition (2013)
6. Mark Gardener, Beginning R: The Statistical Programming Language, Wiley (2013)
7. Michael Lawrence et. al. Programming Graphical User Interfaces in R, Chapman and Hall/CRC (2012)
8. Dr. Rob Kabacoff, R in Action: Data Analysis and Graphics with R, Manning (2015)

(b) Open source software and website address:

Course Name: IoT Application Development	
Semester V	
Course Code:	BCA-EC505
Course Title:	IoT Application Development
Number of Credits:	Total: (L: 2 T:1 P:2)
Total Number of teaching hours	40 Hours
Pre-requisites	
Course Category:	Core

Rationale:

The purpose of this course is to impart knowledge on IoT Architecture and various protocols, study their implementations and various applications of IoT.

Learning Outcome:

- Exemplify the IoT concepts, Physical Design, Logical Design, Enabling Technologies, IoT Challenges, IoT Levels, Applications, IoT Components, Layers and Protocols, Use cloud for IoT applications. Interface embedded boards & program them Develop IoT applications

Detailed Contents

Unit	Title	Hours
Unit 1	IOT Introduction	8
1	Introduction to IoT: Definition & Characteristics of IoT, History and Evolution of IoT, IoT-An Architectural Overview Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. M2M and IoT Technology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics, Knowledge Management	
Unit 2	IOT Data Communication	8
2	Introduction Basics of embedded systems design Embedded OS - Design constraints for mobile applications, both hardware and software related Architecting mobile applications user interfaces for mobile applications touch events and gestures Achieving quality constraints performance, usability, security, availability and modifiability.	
Unit 3	IoT Mobile Apps	8
3	IoT Mobile App Development Trends In 2020 - Role of Mobile Apps in revolutionizing the world of IoT - UX / UI design for IoT Mobile apps - challenges of UX/UI design for IoT applications - practice tips on design for IoT mobile apps IoT App Design Solutions	

Unit 4	Physical Devices and Endpoints	8
4	Physical Devices and Endpoints: Arduino Pin diagram, Arduino Architecture, Arduino Programming, Raspberry Pi Pin diagram, Raspberry Pi Architecture, Compatible Peripherals, Add-Ons, Accessories & Interfaces, Operating System for Raspberry Pi, Setting up and Initial Configuration for Raspberry Pi, Sensors and Interfacing: Controlling LED using switch, Types of Sensors, Integrating Sensors: HDT (Humidity and Temperature Sensor), Light Sensor (LDR), Gas Detector, Ultrasonic Sensor, IR Obstacle Sensor, Fire Sensor and others, Wi-Fi Module, Bluetooth Module, GSM, Camera, Intel Galileo Pin Diagram, Architecture and Interfacing and programming.	

List of Practical

1. To study Arduino, Raspberry Pi and Intel Galileo board
2. To configure the boards for interfacing with sensors.
3. Interface, control & program LED using switch.
4. Interfacing & programming Light Sensor
5. Interfacing & programming Humidity and Temperature Sensor
6. Interfacing & programming Gas Detector
7. Interfacing & programming Ultrasonic Sensor
8. Interfacing & programming Obstacle Sensor
9. Interfacing & programming Fire Sensor
10. Interfacing & programming Bluetooth module
11. Interfacing & programming Wi-Fi Module
12. Interfacing & programming GSM
13. Interfacing & programming Camera
14. Web server Creation
15. Computer App Designing with the boards.

Connecting Embedded Boards with Cloud.

Learning Approach:

- To Study the Concepts of IoT & Develop IoT Applications.

References/suggested learning resources:

TextBook and Learning Resource:

- S. K. Vasudevan, A. S. Nagarajan, RMD Sundaram, "Internet of Things", Wiley.
- G. C. Hillar, "Internet of Things with Python", PACKT Publications.
- A. Bahga, V. Madiseti, "Internet of Things: A Hands-on Approach", Orient Blackswan Private Limited - New Delhi
- J. C. Shovic, "Raspberry Pi IoT Projects: Prototyping Experiments for Makers", Apress.
- M. Schwartz, "Internet of things with the Arduino Yun", Packt Publishing Ltd.
- Richard Grimmett, "Intel Galileo Essentials", Packt Publishing.
- O. Hersent, D. Boswarthick, O. Elloumi, "The Internet of Things: Key Applications and Protocols", Wiley.
- Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014.
- Peter Waher, "Learning Internet of Things", PACKT publishing, BIRMINGHAM – MUMBAI
- Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer

Reference Books:

- Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving

- World of M2M Communications”, ISBN: 978-1-118-47347-4, Willy Publications
- Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on Approach)”, 1st Edition, VPT, 2014.
 - C. Dierbach, “Introduction to Computer Science using Python: A Computational Problem-Solving Focus”, John Wiley & Sons.

(b) Open source software and website address:

- http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html
- Swayam Course on Introduction To Internet Of Things, By Prof. Sudip Misra, IIT Kharagpur
- Swayam Course on Introduction To Industry 4.0 And Industrial Internet Of Things By Prof. Sudip Misra, IIT Kharagpur

Course Name: Introduction to DevOps Technology	
Semester V	
Course Code:	BCA-EC506
Course Title:	Introduction to DevOps Technology
Number of Credits:	Total: (L: 2 T:1 P: 2)
Total Number of teaching hours	40 Hours
Pre-requisites	NIL
Course Category:	Elective

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:

- This course covers best practices, tools, and techniques for automating tasks and collaborating with other engineers on projects.

Detailed Contents

Unit	Title	Hours
Unit 1	Overview of DevOps	10
1.1	Why DevOps, What is DevOps, DevOps Market Trends DevOps Engineer Skills, DevOps Delivery Pipeline, DevOps Ecosystem <i>/ Arch.</i>	
1.2	What is <u>version control</u> , What is Git, Why Git for your organization, Install Git, Common <u>commands in Git</u> , Working with Remote Repositories <i>Life cycle</i>	
Unit 2	Git, Jenkins & Maven Integration & Continuous Integration using Jenkins	10

2.1	Branching and Merging in Git, <u>Git workflows</u> , Git cheat sheet, <u>What is CI</u> , <u>Why CI is Required</u> , Introduction to <u>Jenkins</u> (With <u>Architecture</u>), Introduction to Maven	
2.2	Jenkins Management, <u>Adding a slave node to Jenkins</u> , Building Delivery Pipeline, Pipeline as a Code, Implementation of Jenkins, Build the pipeline of jobs using Jenkins, <u>Create a pipeline script to deploy an application over the tomcat server</u>	
Unit 3	Continuous Deployment: Containerization with Docker	10
3.1	Introducing Docker, Understanding <u>images and containers</u> , Running Hello World in Docker, Introduction to Container, <u>Container Life Cycle</u> , Sharing and Copying Base Image, <u>Dockerfile</u> Working with containers, <u>Publishing Image on Docker Hub</u>	
3.2	Introduction to Docker Ecosystem, <u>Docker Compose</u> , <u>Docker Swarm</u> , Managing Containers, Running Containers, Introduction to <u>Docker Networking</u> , Network Types, Docker Container Networking	
Unit 4	Introduction to DevOps on Cloud	10
4.1	Why Cloud, Introduction to <u>Cloud Computing</u> , <u>Why DevOps on Cloud</u> , Introduction to AWS Various, <u>AWS services</u> , DevOps using AWS	

List of Practical:

As per the Subject teacher assigned as per theory curriculum.

Learning Approach:

The goal of this syllabus is to give you an overview of the skills needed for success in this role so that when a student is looking for work or interviewing for positions, he will be able to confidently say Yes, I know how to do that.

1. (a) Books

1. *The DevOps Handbook: How to Create World-Class Agility, Reliability & Security in Technology Organizations*, written by Gene Kim, Jez Humble, Patrick Debois, and John Willis, was published by IT Revolution Press in 2016.

2. *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale* by Jennifer Davis and Ryn Daniels was also released in 2016. It was published by O'Reilly Media, Inc.
3. *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise* by Sanjeev Sharma was released in 2017. It was published by Wiley.



