Gurugram University, Gurugram

GENERAL COURSE STRUCTURE AND CREDIT DISTRIBUTION

CREDIT DISTRIBUTION

UNDERGRADUATE PROGRAMME

Scheme of Bachelor of Computer Applications Programme

(Scheme UG A4: Undergraduate Programmes (Interdisciplinary))

Semester 1

Course Code	Course Title	CourseID	L	T	P	L	T	P	Total Credits	MARKS				
			(Hrs)			Credits				TI	TE	PI	PE	Total
Core Course(s)														
CC-ID1	Fundamentals of Computers & Programming in C	240/BCA/CC101	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID2	PC Software	240/BCA/CC102	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID3	Logical Organization of Computers	240/BCA/CC103	3	1	0	3	1	0	4	30	70	--	--	100
Minor/ Vocational Course(s)														
MIC-1	One from the pool	240/BCA/MI101							2					50
Multidisciplinary Course(s)														
MDC-1	One from the pool	240/BCA/MD101	3	-	-	3	-	-	3	25	50	--	--	75
Ability Enhancement Course(s)														
AEC-1	One from the pool	240/BCA/AE101	1	1	-	1	1	-	2	15	35	--	--	50
Skill Enhancement Course(s)														
SEC-1	One from the pool	240/BCA/SE101							3					75
Value-added Course(s)														
VAC-1	One from the pool	240/BCA/VA101	2	-	-	2	-	-	2	15	35	--	--	50
Total Credits									**24**					**600**

Semester 2

Course Code	Course Title	CourseID	L	T	P	L	T	P	Credits	MARKS				
			(Hrs)			Credits				TI	TE	PI	PE	Total
Core Course(s)														
CC-ID4	Data Structures	240/BCA/CC201	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID5	Mathematical foundations of Computer Systems	240/BCA/CC202	3	1	-	3	1	-	4	30	70	--	--	100
CC-ID6	Database Management Systems	240/BCA/CC203	3	-	2	3	-	1	4	25	50	05	20	100
Minor/ Vocational Course(s)														
MIC-2	One from the pool	240/BCA/MI201							2					50
Multidisciplinary Course(s)														
MDC-2	One from the pool	240/BCA/MD201	3	-	-	3	-	-	3	25	50	--	--	75
Ability Enhancement Course(s)														
AEC-2	One from the pool	240/BCA/AE201	1	1	-	1	1	-	2	15	35	--	--	50
Skill Enhancement Course(s)														
SEC-2	One from the pool	240/BCA/SE201							3					75
Value-added Course(s)														
VAC-2	One from the pool	240/BCA/VA201	2	-	-	2	-	-	2	15	35	--	--	50
Total Credits									**24**					**600**

Semester 3

Course Code	Course Title	CourseID	L	T	P	L	T	P	Credits	MARKS				
			(Hrs)			Credits				TI	TE	PI	PE	Total
Core Course(s)														
CC-ID7	Operating Systems	240/BCA/CC301	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID8	Object oriented Programming using C++	240/BCA/CC302	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID9	Web Designing	240/BCA/CC303	2	-	2	2	-	1	3	15	35	05	20	75
Minor/ Vocational Course(s)														

MIC-3	One from the pool	240/BCA/MI301							4					100
Multidisciplinary Course(s)														
MDC-3	One from the pool	240/BCA/MD301	3	-	-	3	-	-	3	25	50	--	-	75
Ability Enhancement Course(s)														
AEC-3	One from the pool	240/BCA/AE301	1	1	-	1	1	-	2	15	35	--	--	50
Total Credits									**20**					**500**

Semester 4

Course Code	**Course Title**	**CourseID**	**L**	**T**	**P**	**L**	**T**	**P**	**Credits**	**MARKS**				
			(Hrs)			**Credits**				**TI**	**TE**	**PI**	**PE**	**Total**
Core Course(s)														
CC-ID10	Computer Networks	240/BCA/CC401	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID11	Software Engineering	240/BCA/CC402	3	1	-	3	1	-	4	30	70	--	--	100
CC-ID12	Java Programming	240/BCA/CC403	3	-	2	3	-	1	4	25	50	05	20	100
Minor/ Vocational Course(s)														
MIC-4	One from the pool	240/BCA/MI401							4					100
Ability Enhancement Course(s)														
AEC-4	One from the pool	240/BCA/AE401	1	1	-	1	1	-	2	15	35	--	--	50
Value-added Course(s)														
VAC-3	One from the pool	240/BCA/VA401	2	-	-	2	-	-	2	15	35	--	--	50
Total Credits									**20**					**500**

Semester 5

Course Code	**Course Title**	**Course ID**	**L**	**T**	**P**	**L**	**T**	**P**	**Credits**	**MARKS**				
			(Hrs)			**Credits**				**TI**	**TE**	**PI**	**PE**	**Total**
Core Course(s)														
CC-ID13	Computer Graphics	240/BCA/CC501	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID14	Python Programming	240/BCA/CC502	3	-	2	3	-	1	4	25	50	05	20	100

CC-ID15	Analysis and design of algorithm	240/BCA/CC503	3	-	2	3	-	1	4	25	50	05	20	100
Minor/ Vocational Course(s)														
VOC-1	One from the pool	240/BCA/MI501							4					100
Skill Enhancement Course(s)														
Internship		240/BCA/INT501							4					100
Total Credits									**20**					**500**

Semester 6

Course Code	**Course Title**	**Course ID**	**L**	**T**	**P**	**L**	**T**	**P**	**Credits**	**MARKS**				
			(Hrs)			**Credits**				**TI**	**TE**	**PI**	**PE**	**Total**
Core Course(s)														
CC-ID16	Management Information Systems	240/BCA/CC601	3	1	-	3	1	-	4	30	70	--	--	100
CC-ID17	Compiler Design	240/BCA/CC602	3	-	-	3	-	-	3	25	50	--	--	75
CC-ID18	Artificial Intelligence	240/BCA/CC603	3	-	2	3	-	1	4	25	50	05	20	100
Minor/ Vocational Course(s)														
VOC-2	One from the pool	240/BCA/VO601							4					100
VOC-3	One from the pool	240/BCA/VO602							4					100
Skill Enhancement Course(s)														
SEC-3	One from the pool	240/BCA/SE601							3					75
Total Credits									**22**					**550**

Note: *The curriculum of semester 7 and 8 will is provided in due course of time.*

Semester 1

Course code	CC-ID1			
Category	Core Courses (CC)			
Course title	Fundamentals of Computer & Programming in C			
Course ID	240/BCA/CC101			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course outcomes

CO1: Learn the functional units and classify types of computers, their applications and effects.

CO2: Understand system softwares and their working.

CO3: Understand the logic building used in programming.

CO4: Design and develop algorithms for solving various real-life problems.

UNIT-I

Computer Fundamentals: Concept of data and information. Components of Computer. Input and Output Device, Components of CPU, Memory and Storage Devices, Classification of Computers, Advantages and Limitations of Computer, Applications of Computer, Social concerns of Computer Technology: Positive and Negative Impacts, Computer Crimes, Viruses and their remedial solutions.

Computer Software: System and Application Software, Overview of Operating System Programming Languages Machine. Assembly. High Level Language, 4GL. Language Translator, Linker and Loader.

UNIT-II

Problem Solving: Problem Identification. Analysis, Algorithms, Flowcharts. Pseudo codes. Decision Tables. Program Coding. Program Testing and Execution.

C Programming Fundamentals: Keywords, Variables and Constants, Structure of a C program.

UNIT-Ill

Operators & Expressions: Arithmetic, Unary, Logical. Bit-wise, Assignment & Conditional Operators.

Decision Making: Decision making using if...else. Else If Ladder; Switch, break. Continue and Goto statements.

UNIT-IV

Loops: Looping using while, do...while, for statements. Nested loops.

Functions: Defining & Accessing User defined functions. Library Functions, Function Prototype, Passing Arguments, Passing array as argument. Recursion, Use of Library Functions. Macro vs. Functions, Pointers in C.

Textbooks & Reference Books:

1. Nasib Singh Gill, Computing Fundamentals and Programming in C, Khanna Books Publishing Co., New Delhi.
2. Rajender Singh Chhillar, Application of IT to Business, Ramesh Publishers, Jaipur.
3. E. Balaguruswamy, Programming in C. Tata McGraw Hill.
4. Kanetker Yashwant, Let us C, BPB.
5. Rajaraman, V., Computer Programming in C, PHI

Fundamentals of Computer & Programming in C LAB

List of Experiments

1. Write a C program to compute roots of quadratic equation $ax^2+bx+c=0$, where a, b, and c are three coefficients of a quadratic equation are inputs.

2. Design and develop an algorithm to find the reverse of an integer number.

3. Design and develop an algorithm to check whether given number is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: 2019, Reverse: 9102, Not a Palindrome.

4. Design and develop a c program to implement simple calculator using switch case statement.

5. Develop, implement and execute a C program to search a Number in a list using linear searching Technique.

6. Develop an algorithm, implement and execute a C program that reads N integer numbers and arrange them in ascending order using Bubble Sort.

7. Design and develop a C program to read and print a matrix and check whether a given Matrix is a sparse Matrix or not.

8. a. Write a C program to implements the following string manipulation functions till the use wishes to continue (infinite loop): (i) strcpy() (ii) srrlen() (iii) strrev () (iv) strcmp() (v) strcat().

   b. Read a sentence and print frequency of vowels and total count of consonant

9. Design and develop a C function RightRotate (x, n) that takes two integers x and n as input and returns value of the integer x rotated to the right by n positions. Assume the integers are unsigned.

10. Draw the flowchart and write a recursive C function to find the factorial of a number, n!, define by fact(n)=1, if n=0. Otherwise fact (n) =n*fact (n-1). Using this function, write a C program to compute the binomial coefficient nCr. Tabulate the results for different values of n and r with suitable messages

11. Write a C program to maintain a record of n student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Assume appropriate data type for each field. Input & Print the members of the structure

12. Write a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

Course code	CC-ID2			
Category	Core Courses (CC)			
Course title	PC Software			
Course ID	240/BCA/CC102			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course outcomes

CO1: Understand the basics of Windows operating system including its history, components, and user interface elements.

CO2: learn to create, edit, format, and manage documents using MS Word and apply advanced features such as Mail Merge and Tables for efficient document creation and management.

CO3: Implement data analysis techniques such as sorting, filtering, and conditional formatting in MS Excel.

CO4: Develop presentations using MS PowerPoint, including slide creation, formatting, and layout design.

UNIT – I

Windows: Basics of Windows. Windows History, Basic components of windows, icons, types of icons, taskbar, activating windows, using desktop, title bar, running applications, Windows explorer, computer, managing files and folders, Configuring System devices. Control panel, using windows accessories.

UNIT – II

Documentation Using Word-Processing package: Introduction to Word-Processing, Creating & Editing Document, Formatting Document, Auto-text, Autocorrect, Spelling and Grammar Tool, Document Dictionary, Page Formatting, Bookmark, Advance Features of Word-Processing: Mail Merge, Macros, Tables, Printing, Styles, linking and embedding object.

UNIT – III

Electronic Spread Sheet - Introduction & area of use, Creating & Editing Worksheet, Formatting and Essential Operations, Formulas and Functions, Charts, Database Management : Sorting, Querying, Filtering, Table, Validation, Goal Seek, Scenario.

UNIT – IV

Presentation using PowerPoint: Presentations, Creating, Manipulating & Enhancing Slides, Organizational Charts, Excel Charts, Word Art, Layering art Objects, Animations and Sounds, Inserting Animated Pictures or Accessing through Object, Inserting Recorded Sound Effect or In-Built Sound Effect.

Textbooks & Reference Books:

1. Microsoft Office Complete Reference, BPB Publication
2. Russell A. Stultz Learn Microsoft Office, BPB Publication
3. Courter, G Marquis (1999). Microsoft Office 2000: Professional Edition. BPB.
4. Koers, D (2001). Microsoft Office XP Fast and Easy. PHI.
5. Nelson, S L and Kelly, J (2002). Office XP: The Complete Reference. Tata McGraw Hill.

PC Software LAB

List of Experiments

MS Word:

- Adding text, editing text, finding and replacing text
- Formatting text: font styles, sizes, colors, bold, italic, underline
- Working with styles: creating, modifying, applying styles
- Text indentation: first line, hanging, left and right
- Page layout: setting margins, changing page size, orientation
- Printing a document: adjusting print settings, previewing before print
- Inserting page numbers, headers, footers
- Inserting date and time and Inserting pictures, objects, shapes
- Creating bulleted and numbered lists
- Working with tables: creating, formatting, editing tables
- Working with paragraphs and columns: alignment, spacing
- Reviewing documents: track changes, adding comments, spell check, grammar check
- Mail merge: creating from letters

MS Excel:

- Entering data into cells
- Formatting data: applying borders, currency formats, number formats, fonts
- Creating custom lists
- Using auto fill for data series
- Finding and replacing data

- Editing text: cut, copy, paste, paste special
- Working with formulae: basic arithmetic, using functions
- Applying conditional formatting to highlight data
- Sorting and filtering data: auto filter, advanced filter
- Working with charts: creating 2D charts
- Page layout and printing options: adjusting print area, page setup

MS PowerPoint:

- Creating and formatting slides in a presentation
- Inserting graphics or pictures into slides
- Placing a text box in the title slide with specific content (e.g., name)
- Inserting transitions between slides
- Applying custom animations to text and objects
- Applying transitional effects to slides

Course code	CC-ID3			
Category	Core Courses (CC)			
Course title	Logical Organization of Computers			
Course ID	240/BCA/CC103			
Scheme and Credits	L	T	P	Credits
	3	1	0	4
Theory Internal	30			
Theory External	70			
Total	100			
Duration of Exam	3 hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course outcomes

CO1: Understand number systems, error detecting correcting code and representations of numbers in a computer system.
CO2: Learn computer arithmetic and Boolean algebra and simplification of Boolean expressions.
CO3: Understand working of logic gates and design various combinational circuits using these logic gates.
CO4: Understand working of different types of flip-flops and design different types of registers.

UNIT - I

Information Representation: Number Systems, Binary Arithmetic, Fixed-point and Floating point representation of numbers, BCD Codes, Error detecting and correcting codes, Character Representation – ASCII, EBCDIC, Unicode

Binary Logic: Boolean Algebra, Boolean Theorems, Boolean Functions and Truth Tables, Canonical andStandard forms of Boolean functions, Simplification of Boolean Functions –Venn Diagram, Karnaugh Maps.

UNIT - II

Digital Logic: Basic Gates – AND, OR, NOT, Universal Gates – NAND, NOR, Other Gates – XOR, XNOR etc. Implementations of digital circuits using gates, Combinational Logic – Characteristics, Design Procedures, analysis procedures.

Combinational Circuits: Half-Adder, Full-Adder, Half-Subtractor, Full-Subtractor, Encoders, Decoders, Multiplexers, Demultiplexers, Comparators, Code Converters.

UNIT - III

Sequential Logic: Characteristics, Flip-Flops, Clocked RS, D type, JK, T type and Master Slave flip-flops. State table, state diagram and state equations. Flip-flop excitation tables

Designing registers – Serial Input Serial Output (SISO), Serial Input Parallel Output (SIPO), Parallel Input Serial Output (PISO), Parallel Input Parallel Output (PIPO) and shift registers. Designing counters – Asynchronous and Synchronous Binary Counters, Modulo-N Counters and Up-Down Counters

UNIT - IV

Memory & I/O Devices: Memory Parameters, Semiconductor RAM, ROM, Magnetic and Optical Storage devices, Flash memory, I/O Devices and their controllers.

Instruction Design & I/O Organization: Machine instruction, Instruction set selection, Instruction cycle, Instruction Format and Addressing Modes. I/O Interface, Interrupt structure, Program-controlled, Interrupt-controlled & DMA transfer, IOP.

Textbooks & Reference Books:

1. M. Morris Mano, Digital Logic and Computer Design, Prentice Hall of India Pvt. Ltd.
2. V. Rajaraman, T. Radhakrishnan, An Introduction to Digital Computer Design, Prentice
3. Hall of India Pvt. Ltd.
4. Andrew S. Tanenbaum, Structured Computer Organization, Prentice Hall of India Pvt. Ltd.
5. Nicholas Carter, Schaum's Outlines Computer Architecture, Tata McGraw-Hill

SEMESTER 2

Course code	**CC-ID4**			
Category	Core Subject			
Course title	**Data Structure**			
Course ID	240/BCA/CC201			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course Outcomes:

CO1: Choose appropriate data structures and algorithms and use it to design solution for a specific problem.

CO2: Execute the operations of hashing to retrieve data from data structure.

CO3: Comprehend and select algorithm design approaches in a problem specific manner.

CO4: Design and analyze programming problem statements.

UNIT – I

Data Structure Definition, Data Type vs. Data Structure, Categories of Data Structures, Data Structure Operations, Applications of Data Structures, Algorithms Complexity and Time-Space Trade-off, Big-O Notation. Strings: Introduction, Strings, String Operations, Pattern Matching Algorithms.

UNIT – II

Arrays: Introduction, Linear Arrays, Representation of Linear Array in Memory, Traversal, Insertions, Deletion in an Array, Multidimensional Arrays, Sparse Matrix. Algorithm for Insertion, Deletion Addition and Multiplication in 2- D Array. Searching and Sorting Techniques, Sorting Techniques: Bubble Sort, Merge Sort, Selection Sort’, Heap Sort, Insertion Sort. Searching Techniques: Sequential Searching, Binary Searching, Search Trees.

UNIT – III

Stacks & Queues: Representation of Stacks, Stack Operations, Applications, Queues, Operations on Queues, Circular Queues, Dequeue, Priority Queues, Applications. Linked Lists: Introduction, Types, Operations (Insertion, Deletion, Traversal, Searching, Sorting), Applications, 16 Dynamic Memory Management, Implementation of Linked Representations.

UNIT – IV

Trees: Basic Terminology, Representation, Binary Trees, Tree Representations using Linked List, Basic Operation on Binary tree, Traversal of Binary Trees: In order, Pre-order & Post-order, Applications of Binary tree. Algorithm of Tree Traversal with and without Recursion. Graphs: Definitions and Basic Terminologies, Representation of Graphs, Graph Traversals, Shortest Path Problem, Applications.

Textbooks & Reference Books:

1. Seymour Lipschutz, Data Structures, Tata McGraw-Hill Publishing Company Limited, Schaum's Outlines.
2. YedidyanLangsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, Data Structures Using C, Pearson Education.
3. Trembley, J.P. And Sorenson P.G., An Introduction to Data Structures With Applications, Mcgraw-Hill.
4. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Addison-Wesley.

Data Structure Lab

List of Experiments

1. Write a program to search an element in a two-dimensional array using linear search.

2. Using iteration & recursion concepts write programs for finding the element in the array using Binary Search Method

3. Write a program to perform following operations on tables using functions only (a) Addition (b) Subtraction (c) Multiplication (d) Transpose

4. Using iteration & recursion concepts write the programs for Quick Sort Technique

5. Write a program to implement the various operations on string such as length of string concatenation, reverse of a string & copy of a string to another.

6. Write a program for swapping of two numbers using 'call by value' and 'call by reference strategies.

7. Write a program to implement binary search tree.

8. (Insertion and Deletion in Binary search Tree)

9. Write a program to create a linked list & perform operations such as insert, delete, update, reverse in the link list

10. Write the program for implementation of a file and performing operations such as insert, delete, update a record in the file.

11. Create a linked list and perform the following operations on it (a) add a node (b) Delete a node

Course code	**CC-ID5**			
Category	Core Subject			
Course title	**MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE**			
Course ID	240/BCA/CC202			
Scheme and Credits	L	T	P	Credits
	3	1	0	4
Theory Internal	30			
Theory External	70			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course Outcomes:

CO1: Gain proficiency in measures of central tendency and dispersion, correlation, and regression analysis, enabling effective data analysis and interpretation.

CO2: Understand fundamental algorithms such as exponentiation, linear search, and binary search, and grasp concepts in graph theory including types of graphs, adjacency matrices, and Eulerian and Hamiltonian paths.

CO3: Master concepts related to trees including minimum distance and spanning trees, and algorithms like merge sort, insertion sort, and bubble sort, enhancing problem-solving skills in algorithmic design.

CO3: Develop proficiency in solving recurrence relations and understanding their applications, and grasp foundational concepts in number theory including mathematical induction, GCD, Euclidean algorithm, and public key encryption schemes.

UNIT-I

Basic Statistics: Measure of Central Tendency, Preparing frequency distribution table, Mean, Mode, Median, Measure of Dispersion: Range, Variance and Standard Deviations, Correlation and Regression.

UNIT-II

Algorithm: Algorithms, merits and demerits, Exponentiation, How to compute fast exponentiation. Linear Search, Binary Search, "Big Oh" notation, Worst case, Advantageof logarithmic algorithms over linear algorithms, complexity.

Graph Theory: Graphs, Types of graphs, degree of vertex, sub graph, isomorphic and homeomorphic graphs, Adjacent and incidence matrices, Path Circuit ; Eulerian, Hamiltonian path circuit.

UNIT-III

Tree: Trees, Minimum distance trees, Minimum weight and Minimum distance spanningtrees.

Recursion: Recursively defined function.

Merge sort, Insertion sort, Bubble sort, and Decimal to Binary.

UNIT-IV

Recurrence Relations: LHRR, LHRRWCCs, DCRR. Recursive procedures.

Number Theory: Principle of Mathematical induction**,** GCD, Euclidean algorithm, Fibonacci numbers, congruences and equivalence relations, public key encryption schemes.

Textbooks & Reference Books:

1. Gupta S.P. and Kapoor, V.K., Fundamentals of Applied statistics, Sultan Chand & Sons, 1996.
2. Gupta S.P. and Kapoor, V.K., Fundamentals of Mathematical statistics, Sultan Chand and Sons, 1995.
3. Graybill, Introduction to Statistics, McGraw.
4. Anderson, Statistical Modelling, McGraw.
5. Babu Ram : Discrete Mathematics

Course code	**CC-ID6**			
Category	Core Subject			
Course title	Database Management System			
Course ID	240/BCA/CC203			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course Outcomes:

CO1: Express the basic concepts of DBMS and RDBMS.

CO2: Apply normalization theory to the normalization of a database

CO3: Apply the concept of Transaction Management & Recovery techniques in RDBMS.

CO4: Analyze various concurrency control and recovery techniques.

UNIT-I

Database System Concepts and Architecture: Traditional File Processing System vs DBMS, Characteristics & Advantages of DBMS, Three-Schema Architecture and Data Independence; Data Models, Schemas, and Instances; Database Languages and Interfaces; Classification of DBMS.

Data Modeling: Overview of Entity-Relationship Diagram, Relational Model -Constraints, Relational Database Schemas, Relational Algebra and Relational Calculus; Codd Rules.

UNIT-II

Normalization for Relational Databases: Functional Dependencies and Normalization; **SQL:** SQL as 4GL, SQL Components: DDL, DML, DQL, DCL, TCL; Data Definition and Data Types; Constraints, Queries, Insert, Delete, and Update Statements; Views, Stored Procedures and Functions; Database Triggers, SQL Injection.

UNIT-Ill

Query Processing and Optimization: Translating SQL queries into Relational Algebra, Basic Algorithm for Executing Query Operations, Using Heuristic in Query Optimization, Using Selectivity and Cost Estimation in Query Optimization, Semantic Query Optimization.

Transaction Processing: Introduction, Desirable properties of Transactions, Schedules & Recoverability, Serialization of Schedulers, Transaction Support in SQL.

Basics of Database Security and Authorization.

UNIT-IV

Concurrency Control Techniques: Locking techniques for Concurrency Control, Concurrency Control based on Timestamp ordering, Multiversion Concurrency Control Techniques, Validation Currency Control Techniques, Granularity of data items and multiple granularity locking, Using locks for Concurrency Control in Indexes.

Database Recovery Techniques: Basic Concepts, Recovery Technique based on Deferred Update, Recovery Technique based on Immediate Update, Shadow Paging, The ARIES recovery algorithm, Database backup and recovery from catastrophic failure.

Textbooks & Reference Books:

1. Elmasri & Navathe: Fundamentals of Database Systems, Sth edition, Pearson Education.
2. C.J.Date: An Introduction to DatabaseSystems, 8thedition, Addison Wesley N. Delhi.
3. Thomas Connolly, CarolynBegg: Database Systems, Pearson Education.

Data Base Management System Lab

List of Experiments

1. Creation of a database and writing SQL queries to retrieve information from the database.
2. Performing Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.
3. Creation of Views, synonyms, Sequences, indexes, and save points.
4. Creating an employee database to set various constraints.
5. Mini project (Any 2 Application Development using MySQL)
    a. Inventory Control System

b. Material Requirement Processing.
c. Hospital Management System.
d. Railway Reservation System.
e. Personal Information System.
f. Web-Based User Identification System.
g. Time Table Management System.
h. Hotel Management

SEMESTER 3

Course code	CC-ID7			
Category	Core Course			
Course title	**Operating System**			
Course ID	240/BCA/CC301			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1: understand the basic concepts of operating systems and its services.

CO2: understand concept of process management and scheduling.

CO3: acquire knowledge of process synchronization along with deadlock handling.

CO4: learn about memory management and distributed operating system.

UNIT – I

Introductory Concepts: Operating System Functions and Characteristics, Historical Evolution of Operating Systems, Operating System Structure and Operations; Types of Operating System: Real time, Multiprogramming, Multiprocessing, Batch processing; Operating System Services, Operating System Interface, Methodologies for Implementation of Operating System, Service System Calls, System Programs.

UNIT – II

Process Management: Process Concepts, Operations on Processes, Process States and Process Control Block. InterProcess Communication; Multithreaded Programming: Multithreading Models, Threading Issues; CPU Scheduling: Scheduling Criteria, Levels of Scheduling, Scheduling Algorithms, Multiple Processor Scheduling; Algorithm Evaluation.

UNIT – III

Synchronization: Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problem of Synchronization, Monitors, Atomic Transactions; Deadlocks: Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock 20 Avoidance, Deadlock Detection and Recovery.

UNIT – IV

Memory Management Strategies: Memory Management of Single-User and Multiuser Operating System, Partitioning, Swapping, Contiguous Memory Allocation, Paging and Segmentation; Virtual Memory Management: Demand Paging, Page Replacement Algorithms, Thrashing, Memory Mapped Files. Distributed Operating Systems: Types of Network based Operating Systems, Network Structure, Design Issues;

Textbooks & Reference Books:

1. Silberschatz A., Galvin P.B.,and Gagne G., Operating System Concepts, John Wiley & Sons.
2. Godbole, A.S., Operating Systems, Tata McGraw-Hill Publishing Company, New Delhi.
3. Deitel, H.M., Operating Systems, Addison-Wesley Publishing Company, New York.
4. Tanenbaum, A.S., Operating System-Design and Implementation, Prentice Hall of India, New Delhi.

Operating System Lab

List of Experiments

1. Implementation of FCFS and SJF CPU scheduling algorithms
2. Implementation of Round Robin and Priority CPU Scheduling
3. Implementation of Producer-Consumer problem using semaphores
4. Implementation of Dining Philosphers Problem
5. Implementation of FIFO Page Replacement Algorithms
6. Implementation of LRU Page Replacement Algorithms
7. Implementation of Sequential File Allocation Strategies
8. Implementation of Indexed File Allocation Strategies

Course code	**CC-ID8**			
Category	Core course			
Course title	**Object Oriented Programming Using C++**			
Course ID	240/BCA/CC302			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1: understand pointers, constructor and destructors in C++;

CO2: acquire the detailed knowledge of polymorphism;

CO3: learn to implement exception handling and template;

CO4: learn File handling in C++.

UNIT-I

Elements of Programming and Function Introduction: Basic Elements of Programming, Console I/O Operations, Function: Function Prototyping, Call and Return By Reference, Inline Function, Default and Const Arguments, Function Overloading, Arrays, Manipulators and Enumeration.

UNIT-II

Classes and Object Oriented Methodology: Basic Concepts/Characteristics of OOP. Advantages and Application of Oops, Procedural Programming Vs OOP. Classes and Objects: Specifying a Class, Creating Objects, Private & Public Data Members and Member Functions, Defining Inline

Member Functions, Static Data Members and Member Functions. Arrays within Class, Arrays of Objects, Objects as Function Arguments, Returning Objects.

UNIT-III

Constructors, Destructors, Operators Overloading and Inheritance. Constructors and Destructors: Introduction, Parameterized Constructors, Multiple Constructors in A Class, Constructors With Default Arguments, Dynamic Initialization of Objects, Copy Constructors, Dynamic Constructors, Const Objects, Destructors Operators Overloading: Definition, Unary and Binary Overloading, Rules for Operator Overloading. Inheritance: Defining Derived Classes, Types of Inheritance, Constructors and Destructors in Derived Classes.

UNIT-IV

Pointers Virtual & Friend functions and file handling Pointers: Pointer to Objects, This Pointer, "New" and "Delete"Operators, Virtual Function, Friend Functions. Opening, Closing A File, File Modes, File Pointers and Their Manipulation, Sequential Input and Output Operations: Updating A File, Random Access, and Error Handling During File Operations, Command Line Arguments.

Text and Reference Books:

1. K.R.Venugopal, Rajkumar, T. Ravishankar, "Mastering C++", TMH ,ISBN:0-07- 463454-2.
2. Farrel,"Object-Oriented Programming using C++",Cenage Pub, ISBN: 9788131505175
3. Parimala N.," Object Orientation through C++", Macmillan India Ltd. Publication, ISBN:-0333 93202-1.
4. E Balagurusamy, "Object Oriented Programming with C++ ", Tata McGraw Hill Publishing Company Limited, New Delhi, ISBN:- 13- 978-07-066907-9.

Object Oriented Programming Using C++ LAB

List of Experiments

1. C ++ program to find first 10 prime numbers
2. C++ program to implement Constructor Overloading
3. C++ program Invocation order of Constructor and Destructor in Inheritance
4. C++ program to find area of Triangle, Circle and Rectangle
5. C++ program to implement Static Variable and Function
6. C++ program to implement Inheritance

7. C++ program to implement Multiple Inheritance
8. C++ program to implement Function Overloading
9. C++ program to implement Operator Overloading
10. C++ program to implement constructor, destructor and scope resolution operator

Course code	**CC-ID9**			
Category	Core Course			
Course title	**Web Designing**			
Course ID	240/BCA/CC303			
Scheme and Credits	L	T	P	Credits
	2	0	2	3
Theory Internal	15			
Theory External	35			
Practical Internal	05			
Practical External	20			
Total	75			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1: outline the idea of web and its components.

CO2: understand the theoretical steps for developing a website.

CO3: learn the skills that will enable him/her to design simple web pages.

CO4: learn CSS to specify style to web pages.

UNIT – I

Introduction to Internet and World Wide Web (WWW); Evolution and History of World Wide Web, Web Pages and Contents, Web Clients, Web Servers, Web Browsers; Hypertext Transfer Protocol, URLs; Searching and WebCasting Techniques, Search Engines and Search Tools, Scripting Languages.

UNIT – II

Web Publishing: Hosting website; Internet Service Provider; Planning and designing website; Web Content Authoring, Web Graphics Design, Web Programming, Steps For Developing website, Choosing the Contents, Home Page, Domain Names, Creating a Website and Introduction to Mark up Languages (HTML and DHTML).

UNIT – III

Web Development: HTML Document Features, Fundamentals HTML Elements, Creating Links; Headers; Text styles; Text Structuring; Text colour and Background; Formatting text; Page layouts, Images; Ordered and Unordered lists; Inserting Graphics; Table Creation and Layouts; Frame Creation and Layouts; Working with Forms and Menus; Working with Radio Buttons; Check Boxes; Text Boxes.

UNIT – IV

Introduction to CSS (Cascading Style Sheets): Features, Core Syntax, Types, Style Sheets and HTML, Style Rule Cascading and Inheritance, Text Properties, CSS Box Model, Normal Flow Box Layout, Positioning and other useful Style Properties; Features of CSS3.

Text and Reference Books:

1. Raj Kamal, Internet and Web Technologies, Tata McGraw-Hill.
2. Ramesh Bangia, Multimedia and Web Technology, Firewall Media.
3. Thomas A. Powell, Web Design: The Complete Reference, Tata McGraw-Hill
4. Wendy Willard, HTML Beginners Guide, Tata McGraw-Hill.
5. Deitel and Goldberg, Internet and World Wide Web, How to Program, PHI

WEB DESIGNING LAB

List of Experiments

1. Write a HTML program to create a webpage about the different art forms of India, with appropriate title on the title bar. Use different heading tags for the headings, and list them using ordered list.
2. Write a HTML program to create sections in the document using appropriate tags and apply different color as background to them. Use internal hyperlinks to move to different points within the page.
3. Write a HTML program to insert a picture on the webpage, giving description for the picture in a paragraph. Use properties of height, width, hspace, vspace and align, with different values.
4. Write a HTML Program, to create a profile of, the First page containing the applicant's picture with personal details using unordered lists, and the second containing Educational details using tables. Use hyperlinks to move to the next page.
5. Using Frames create an Indian Flag and insert the image of chakra in the center.

6. Create a frame like structure based on the given diagram, such that When the first link is clicked, the contents of the first frame is filled with the corresponding information and when the second link is clicked the second frame is filled.
7. Write a program in HTML to demonstrate the concept of Image map, for India map. Map for areas rectangle, Circle and polygon.

SEMESTER 4

Course code	**CC-ID10**			
Category	Core Course			
Course title	**Computer Networks**			
Course ID	240/BCA/CC401			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Explain the functions of the different layers of the OSI Protocol.
CO2: Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs), and Wireless LANs (WLANs) and describe the function of each.
CO3: Identify and connect various connecting components of a computer network.
CO4: Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, and Firewalls using open-source available software and tools.
CO5: outline various models, topologies and devices of Computer Networks.

UNIT - I

Introduction: Data communication, Components, Data Representation, Simplex, Half Duplex, and Full Duplex Transmission, Modulation, Multiplexing, Computer networks, distributed processing, Internet, Topologies, Packet and circuit switching, connectionless and connection- oriented services. Network Models: OSI model and TCP/IP Model Physical Layer – LAN: Ethernet.

UNIT – II

Data Link Layer and Medium Access Sub Layer: MAC Addressing, Framing, Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window Protocol. Medium Access Control: Random access, Controlled Access, and channelization protocols. Network Layer: Logical addressing, classful and classless addressing, subnetting, Ipv4, ICMPv4, ARP, RARP and BOOTP, Ipv6, Ipv6 addressing.

UNIT - III

Network Devices: Repeater, hub, switch, router, and gateway. Routing Algorithms: introduction to routing, Shortest Path Algorithm, Flooding, Hierarchical Routing, Link State, and Distance Vector Routing Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), TCP connection management.

UNIT - IV

Congestion Control, Quality of Service, QoS Improving techniques. Application Layer: Domain Name Space (DNS), EMAIL, File Transfer Protocol (FTP), HTTP, SNMP Network Security: Firewalls, security goals, types of attack, symmetric and asymmetric key ciphers.

TEXT AND REFERENCE BOOKS:

1. Essentials of Computer And Network Technology, Nasib Singh Gill, Khanna Book Publishing Co.(P) Ltd, New Delhi.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. Computer Networks, latest Edition, Andrew S. Tanenbaum, Pearson New International Edition.
4. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
5. Internetworking with TCP/IP, Volume 1, latest Edition Douglas Comer, Prentice Hall of India.
6. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Computer Networks lab

List of Experiments

1. To construct a simple network topology on Packet Tracer.

2. To verify and configure VLAN and VLAN trunk in packet tracer.

3. To construct RJ45 cable.

4. a) To configure simple static routing.

b) To implement Security on interconnecting devices.

5. To configure a Network Topology constitutes Routers and Switches using Packet Tracer.

6 Working with complex network topologies.

7. Mid Term Evaluation 8. To monitor network traffic using Wire Shark

9. To get the MAC or Physical Address of the system Using Address Resolution Protocol.

10. To Configure network using Routing Information Protocol (RIP)

11. To configure network state routing protocol (OSPF).

12. To configure Border Gateway Protocol.

13. To configure Application Layer protocols: DHCP and DNS.

Course code	**CC-ID11**			
Category	Core Course			
Course title	**Software Engineering**			
Course ID	240/BCA/CC402			
Scheme and Credits	L	T	P	Credits
	3	1	0	4
Theory Internal	30			
Theory External	70			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

At the end of this unit, students will demonstrate the ability to:

CO1: Understand the fundamental principles and characteristics of software, including its evolution and the evolving role of software in modern systems.

CO2: Apply basic concepts of Software Design, including Architectural Design, Modularization, Design Structure Charts, Flow Charts, and measures of Coupling and Cohesion.

CO3: Define Software Quality attributes and understand Software Quality Assurance plans and activities, including Software Documentation

CO4: Understand the need for Software Maintenance and differentiate between categories such as Preventive, Corrective, and Perfective Maintenance.

UNIT-I

Introduction: Software and its Characteristics, Evolving Role of Software, Software Product. Software Processes. Software Crisis. Software Engineering Evolution. Principles of Software Engineering. Programming-in-the-small vs. Programming-in-the-large. Software Components. Software Engineering Processes.

Software Life Cycle (SLC) Models: Water-Fall Model. Prototype Model. Spiral Model. Evolutionary Development Models. Iterative Enhancement Models. Object Oriented Models and other latest Models.

Software Requirements: Functional and Non-Functional. User requirements. System requirements. Software Requirements Document - Requirement Engineering Process: Feasibility Studies. Requirement's elicitation and analysis, requirements validation, requirements management.

UNIT-II

Software Design: Basic Concept of Software Design. Architectural Design. Low Level Design: Modularization. Design Structure Charts. Flow Charts. Coupling and Cohesion Measures; Design Strategies: Function Oriented Design. Object Oriented Design. Top• Down and Bottom-Up Design. User Interface Design. Programming practices and Coding standards.

Software Testing: Introduction. Verification vs. Validation. Software Reliability. Levels of Testing. Structural Testing (WhiteBox Testing). Functional Testing (Black Box Testing).

UNIT-III

Software Quality: Attributes, Software Quality Assurance - plans & activities: Software Documentation.

Software Project Management: Project Management activities. Project Estimation. Project planning. Project scheduling.

Software Risk Management: Reactive versus Proactive Risk Strategies. Risk management activities: Software Risks (Risk Identification. Risk Projection. Risk Refinement. Risk Mitigation). Risks Monitoring and Management.

Software Measurement and Metrics: Process Metrics. Project metrics. Estimation-LOC, Halstead's Software Science. Function Point (FP). Cyclomatic Complexity Measures: Software Project Estimation Models- Empirical. Putnam. COCOMO I & II.

UNIT-IV

Software Maintenance: Need for Maintenance. Categories of Maintenance: Preventive. Corrective and Perfective Maintenance. Cost of Maintenance: Software Re-Engineering. Reverse Engineering, Software Documentation.

Software Configuration Management: SCM Activities. Change Control Process. Software Version Control: Software Reuse. Software Evolution.

CASE Computer Aided Software Engineering(CASE).CASE Tools.TEXT BOOK:

Textbooks & References:

1. Software Engineering, Nasib Singh Gill, Khanna Book Publishing Co.(P) Ltd, New Delhi.
2. Software Engineering – A Practitioner's Approach, Roger S. Pressman, 1996, MGH.

3. Fundamentals of software Engineering, Rajib Mall, PHI
4. Software Engineering by Ian Summerville, Pearson Edu, 5th edition, 1999, AW,
5. Software Engineering – David Gustafson, 2002, T.M.H
6. Software Engineering Fundamentals Oxford University, Ali Behforooz and Frederick J. Hudson 1995 JW&S,
7. An Integrated Approach to software engineering by Pankaj jalote , 1991 Narosa

Course code	**CC-ID12**			
Category	Core Course			
Course title	**Java Programming**			
Course ID	240/BCA/CC403			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Identify classes, objects, members of a class and relationships among them for a specific problem.

CO2: Understand and demonstrate the concepts of garbage collection, polymorphism, inheritance etc.

CO3: Do numeric (algebraic) and string-based computation.

CO4: Understand and implement modularity as well as basic error-handling techniques.

CO5: Develop, design and implement small multithreaded programs using Java language.

UNIT - I

Introduction to Java: Evolution of Java, Object Oriented Programming Structure, Overview and characteristics of Java, Java program Compilation and Execution Process, Organization of the Java Virtual Machine, Client side Programming, Platform Independency & Portability, Security, Relation b/w JVM, JRE and JDK, Introduction to JAR format, Naming Conventions, Data types & Type casting, operators, Security Promises of the JVM.

UNIT - II

OOPS Implementation: Classes, Objects, attributes, methods, data encapsulation, reference variables, Constructors, Anonymous block, Method Overloading, Static Data members, Argument Passing Mechanism: Passing primitive arguments, passing objects, Wrapper Classes; This keyword: Referencing instance members, Intra class constructor chaining, Method chaining; Inheritance & code reusability: Extending classes for code reusability, Usage of super keyword, Method Overriding.

UNIT - III

Packages: Defining Package, CLASSPATH, Package naming, Accessibility of Packages, using Package Members.

Interfaces: Implementing Interfaces, Interface and Abstract Classes, Extends and Implements together.

Exceptions Handling: Exception, Handling of Exception, Using try-catch, Catching Multiple Exceptions, Using finally clause, Types of Exceptions, Throwing Exceptions, Writing Exception Subclasses.

UNIT - IV

Multithreading: Introduction, The Main Thread, Java Thread Model, Thread Priorities, Synchronization in Java, Inter thread Communication.

I/O in Java: I/O Basics, Streams and Stream Classes ,The Predefined Streams, Reading from, and Writing to, Console, Reading and Writing Files , The Transient and Volatile Modifiers , Using Instance of Native Methods.

Strings and Characters: Fundamentals of Characters and Strings, The String Class, String Operations, Data Conversion using Value Of ( ) Methods, String Buffer Class and Methods.

TEXT AND REFERENCE BOOKS:

1. E. Balaguruswamy, “Programming with Java”, TMH

2. Horstmann, “Computing Concepts with Java 2 Essentials”, John Wiley.

3. Decker &Hirshfield, “Programming.Java”, Vikas Publication.

4. Patrick Naughton and HerbertzSchidt, “Java-2 the complete Reference”, TMH

5. Sierra & bates, “Head First Java”, O’Reilly.

Java Programming lab

List of Experiments

1. Write a java program to find the Fibonacci series using recursive and non recursive functions
2. Write a java program to multiply two given matrices.
3. Write a java program for Method overloading and Constructor overloading .
4. Write a program to demonstrate execution of static blocks, static variables & static methods.
5. Write a program to display the employee details using Scanner class
6. Write a program for sorting a given list of names in ascending order
7. Write a program to implement single and Multi level inheritance
8. Write a program to implement Hierarchical Inheritance.
9. Write a program to implement method overriding.
10. Write a program to create an abstract class named Shape that contains two integers and an empty method named printArea (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.
11. Write a program to implement Interface.
12. Write a program to implement multiple and Hybrid Inheritance
13. Write a program to create inner classes
14. Write a program to create user defined package and demonstrate various access modifiers.
15. Write a program to demonstrate the use of super and final keywords.
16. Write a program if number is less than 10 and greater than 50 it generate the exception out of range. Else it displays the square of number.
17. Write a program with multiple catch Statements.
18. Write a Program to implement simple Thread by extending Thread class and implementing runnable interface.
19. Write a program that implements a multi-thread application that has three threads

SEMESTER 5

Course code	**CC-ID13**			
Category	Core Course			
Course title	**Computer Graphics**			
Course ID	240/BCA/CC501			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course Outcomes:

CO1: Develop a comprehensive understanding of computer graphics systems, including display devices, input methods, and their application across various domains.

CO2: Master 2D transformations (translation, scaling, rotation, reflection, shear) and viewing techniques, enabling precise manipulation and visualization of graphical objects.

CO3: Gain proficiency in representing 3D objects using polygonal and spline surfaces, and implementing basic illumination models for realistic rendering.

CO4: Acquire advanced skills in 3D transformations (translation, rotation, scaling, reflection, shear) and viewing techniques, facilitating complex 3D scene creation and visualization.

UNIT-I

Graphics Primitives: Introduction to computer graphics, Basics of Graphics systems, Application areas of Computer Graphics, overview of graphics systems, video-display devices, and raster-scan systems, random scan systems, graphics monitors and workstations and input devices.

Output Primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary fill and flood- fill algorithms .

UNIT-II

2-D Geometrical Transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to view- port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT-III

3-D Object Representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon-rendering methods.

UNIT-IV

3-D Geometric Transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D Viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

TEXT AND REFERENCE BOOKS:

1. Donald Hearn and M. Pauline Baker : Computer Graphics, PHI Publications.
2. Plastock : Theory & Problem of Computer Gaphics, Schaum Series.
3. Foley & Van Dam : Fundamentals of Interactive Computer Graphics, Addison-Wesley.
4. Newman : Principles of Interactive Computer Graphics, McGraw Hill.
5. Tosijasu, L.K. : Computer Graphics, Springer-Verleg.

Computer Graphics Lab

List of Experiment

1. Implement algorithms to draw points and lines using DDA (Digital Differential Analyzer) and Bresenham's line drawing algorithm.
2. Practice drawing circles and ellipses using midpoint circle and ellipse algorithms.
3. Implement scan-line polygon fill algorithm to fill simple polygons.

4. Implement boundary fill and flood-fill algorithms to fill regions with different colors.
5. Implement translation, scaling, rotation, reflection, and shear transformations using matrix representations and homogeneous coordinates.
6. Perform composite transformations (e.g., translate then rotate).
7. Implement the Cohen-Sutherland line clipping algorithm to clip lines against a rectangular window.
8. Implement the Sutherland-Hodgman polygon clipping algorithm to clip polygons against a convex clipping window.
9. Implement algorithms for drawing Bezier curves and B-Spline curves.
10. Implement 3-D transformations including translation, rotation, scaling, reflection, and shear.
11. Combine transformations to perform composite transformations on 3-D objects.
12. Implement perspective and orthographic projections for 3-D objects.

Course code	**CC-ID14**			
Category	Core Course			
Course title	**Python Programming**			
Course ID	240/BCA/CC502			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

Course Outcomes:

CO1: Develop proficiency in fundamental programming concepts including variables, data types, control statements (if-else, loops), and file manipulation (reading/writing text and CSV files).

CO2: Master data structures such as lists, tuples, and dictionaries, and understand function design principles including arguments, return values, and managing program complexity.

CO3: Acquire skills in OOP with classes, inheritance, polymorphism, and exception handling, enabling effective data modeling, abstraction, and code reusability.

CO4: Gain practical experience in creating graphical user interfaces (GUIs) using tkinter, including designing simple interfaces with widgets like buttons, labels, and entry fields, and managing layouts and event-driven programming paradigms.

UNIT-I

The concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; comments in the program; understanding error messages; Conditions, boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while); short-circuit (lazy) evaluation; Strings and text files; manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab separated); String manipulations: subscript operator, indexing, slicing a string.

UNIT-II

Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries; Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments.

UNIT-III

Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects; Encapsulation, Information hiding Method, Signature, Classes and Instances, Review of Abstraction, inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; exception handling, try block

UNIT-IV

Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

TEXT AND REFERENCE BOOKS:

1. Phillips, Dusty. Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8. Packt Publishing Ltd, 2018.

2. Steven F. Lott, Mastering Object-Oriented Python - Second Edition, published by Packt.

3. Python Object Oriented Programming Cookbook, published by Packt.

4. Mark Lutz, Programming Python: Powerful Object-Oriented Programming.

5. Irv Kalb, Object-Oriented Python: Master OOP by Building Games and GUIs Kindle Edition

Python Programming Lab

List of Practical:

1. Basic building blocks of a Python program (variables, conditional statements, loops, libraries, functions, errors).

2. Data structures (trees, dictionaries, tuples)

3. Object Oriented programming (classes, objects, inheritance, polymorphism, abstract classes).

4. PyQt for creating graphical user interfaces for interactive programs

5. NumPy (Matrices, vectors, linear algebra)

6. SciPy (Package for numerical computations)

7. Matplotlib (Plotting) 8. Interactive Python (IPython)

Course code	**CC-ID15**			
Category	Core Course			
Course title	**Analysis and design of algorithm**			
Course ID	240/BCA/CC503			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1: Apply the best data structure for designing an algorithm to solve a given problem.

CO2: Evaluate different algorithms with respect to time and space complexity.

CO3: Create algorithms to solve various computational problems.

CO4: Understand different complexity classes.

CO5: Explain the hardness of real world problems with respect to algorithmic efficiency and learning to cope with it.

UNIT I

BASIC CONCEPTS OF ALGORITHMS: Basic Concepts of Algorithms: Notion of Algorithm, Fundamentals of Algorithmic Solving, Important problem types, Fundamentals of the Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical analysis of nonrecursive algorithms. Mathematical analysis of recursive algorithm: recurrence relations, solution of recurrence relations using substitution method

UNIT II

SORTING ALGORITHMS: Brute Force, Divide and Conquer Strategy: Selection sort, Bubble sort, Sequential searching (Linear Search), Brute force string matching, General method, Merge sort, Quick Sort, Binary Search, Strassen's matrix multiplication

UNIT III

GREEDY AND DYNAMIC PROGRAMMING: Greedy Approach and Dynamic Programming: Fractional Knapsack problem, Minimum cost spanning tree: Prim's and Kruskal's algorithm, Single source shortest path problem, Principle of optimality, Multi-stage graph problem, all pair shortest path problem, 0/1 Knapsack problem, Traveling salesperson problem

UNIT IV

BACKTRACKING & BRANCH AND BOUND: Backtracking and Branch and Bound: General method backtracking, N-Queen problem, 0/1 Knapsack problem, General method of branch & bound, 0/1 Knapsack problem, Traveling sales person problem Lower Bound Theory and Complexity Classes, Lower bounds, Decision trees, P, NP and NP Complete problems

TEXT AND REFERENCE BOOKS:

1. Algorithm Design, Jon Kelinberg and Eva Tardos, 1st Edition, Pearson Education 2014.

2. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Book Publishing 2018.

3. Fundamentals of algorithms, Horowitz E, Sahini S, Rajasekaran S., University Press 2008.

4. Introduction to algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, PHI. 2012

5. An introduction to analysis of algorithms, R. Sedgewick, 1st edition, Pearson Education 1996.

6. Data Structures and Program Design in C By Robert L. Kruse, C.L. Tondo, Bruce Leung, Pearson Education. 2007.

Analysis and Design of Algorithms Lab

List of Practical

1. **Write a program to implement different sorting techniques.**
   - Bubble Sort
   - Insertion Sort
   - Selection Sort
   - Quick Sort
   - Merge Sort
2. Write a program to find minimum cost spanning tress.

3. Write a program to implement travelling sales person problem.
4. Write a program to find Longest Path in a Directed Acyclic Graph.
5. Write a program for Shortest path with exactly k edges in a directed and weighted graph.
6. Write a program find maximum number of edge disjoint paths between two vertices
7. Implement 0/1 Knapsack problem using Dynamic Programming.
8. Perform various tree traversal algorithms for a given tree.
9. Implement N-Queens Problem

SEMESTER 6

Course code	**CC-ID16**			
Category	Core Course			
Course title	**Management Information System**			
Course ID	240/BCA/CC601			
Scheme and Credits	L	T	P	Credits
	3	1	0	4
Theory Internal	30			
Theory External	70			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1: Understand fundamental concepts and principles of information systems and their role in organizational settings.

CO2:Analyze and differentiate between various types of information and their significance in decision-making processes.

CO3: Apply frameworks and models (such as Simon's Model) to analyze structured and unstructured decisions within formal and informal systems.

CO4: Design, develop, implement, and evaluate Management Information Systems (MIS) considering information requirements, levels of management, and potential pitfalls in MIS development.

UNIT – I

Introduction to system and Basic System Concepts, Types of Systems, The Systems Approach, Information System: Definition & Characteristics, Types of information, Role of Information in Decision Making, Sub-Systems of an Information system: EDP and MIS management levels, EDP/MIS/DSS.

UNIT –II

An overview of Management Information System: Definition & Characteristics, Components of MIS, Frame Work for Understanding MIS: Information requirements & Levels of Management, Simon's Model of decision-Making, Structured Vs Un-structured decisions, Formal vs. Informal systems.

UNIT – III

Developing Information Systems: Analysis & Design of Information Systems: Implementation & Evaluation, Pitfalls in MIS Development.

UNIT – IV

Functional MIS: A Study of Personnel, Financial and production MIS, Introduction to e-business systems, ecommerce – technologies, applications, Decision support systems – support systems for planning, control and decision-making

TEXT AND REFERENCE BOOKS:

1. J. Kanter, "Management/Information Systems", PHI.
2. Gordon B. Davis, M. H. Olson, "Management Information Systems – Conceptual foundations, structure and Development", McGraw Hill.
3. James A. O'Brien, "Management Information Systems", Tata McGraw-Hill.
4. James A. Senn, "Analysis & Design of Information Systems", Second edition, McGraw Hill.
5. Robert G. Murdick & Joel E. Ross & James R. Claggett, "Information Systems for Modern Management", PHI.
6. Lucas, "Analysis, Design & Implementation of Information System", McGraw Hill

Course code	**CC-ID17**			
Category	Core Course			
Course title	**Compiler Design**			
Course ID	240/BCA/CC602			
Scheme and Credits	L	T	P	Credits
	3	0	0	3
Theory Internal	25			
Theory External	50			
Total	75			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1: Use compiler construction tools and describes the Functionality of each stage of compilation process

CO2: Construct Grammars for Natural Languages and find the Syntactical Errors/Semantic errors during the compilations using parsing techniques

CO3: Analyze different representations of intermediate code.

CO4: Construct new compiler for new languages.

UNIT – I

INTRODUCTION TO COMPILERS: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyzer, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX-lexical analyzer generator.

UNIT – II

BOTTOM UP PARSING: Definition of bottom up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC-automatic parser generator.

PARSING: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling-else grammar, classes of parsing, top down parsing - backtracking, recursive descent parsing, predictive parsers, LL(1) grammars.

UNIT - III

SYNTAX DIRECTED TRANSLATION: Syntax directed definition, construction of syntax trees, S attributed and L-attributed definitions, translation schemes, emitting a translation.

INTERMEDIATE CODE GENERATION: intermediate forms of source programs– abstract syntax tree, polish notation and three address code, types of three address statements and its implementation, syntax directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

UNIT – IV

CODE OPTIMIZATION: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, global data flow analysis.

CODE GENERATION: Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

TEXT AND REFERENCE BOOKS:

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman (2007), Compilers Principles, Techniques and Tools, 2nd edition, Pearson Education, New Delhi, India.
2. Alfred V. Aho, Jeffrey D. Ullman (2001), Principles of compiler design, Indian student edition, Pearson Education, New Delhi, India.
3. Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.
4. K. L. P Mishra, N. Chandrashekaran (2003), Theory of computer science- Automata Languages and computation, 2nd edition, Prentice Hall of India, New Delhi, India.
5. Andrew W. Appel (2004), Modern Compiler Implementation C, Cambridge University Press, UK.

Course code	**CC-ID18**			
Category	Core Course			
Course title	**Artificial Intelligence**			
Course ID	240/BCA/CC603			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

CO1 - Understand the Basics about Artificial Intelligence and Expert Systems.
CO2 - Understand the Programming Logics in Artificial Intelligence.
CO3 - Understand various search methods in Artificial Intelligence.
CO4 - Understand the Knowledge about the Expert Systems.
CO5 - Understand the latest developments in Knowledge systems and Tools.

UNIT – I

Artificial Intelligence : Intelligence, AI Concepts, Various definitions of AI, Knowledge, Knowledge Pyramid, People and Computers: What computers can do better that people, what people can do better than computers; Characteristics of AI Problems, Problem Representation in AI, Components of AI, AI Evolution, Application Areas of AI, History of AI, The Turing Test, The Revised Turing Test

UNIT – II

Expert System: Components of Expert System: Knowledge Base, Inference Engine, User Interface, Features of Expert System, Expert System Life Cycle, Categories of Expert System, Rule Based vs. Model Based Expert Systems, Advantages/Limitations of Expert System, Developing an Expert System: Identification, Conceptualization, Formalization, Implementation, Testing, Using an Expert System, Application Areas of Expert System

UNIT-III

AI and Search Process: Brute Force Search – Depth First/Breadth First Search, Heuristic Search: Hill Climbing, Constraint Satisfaction, Mean End Analysis, Best First Search, A* Algorithm, AO*Algorithm, Beam Search.

UNIT – III

Natural Language Processing: Introduction, Need, Goal, Fundamental Problems in Natural Language Understanding, How People overcome Natural Language Problems, Speech Recognition: Introduction, Advantages and Approaches, Introduction to Robotics: Parts of a Robot, Controlling a Robot, Intelligent Robots, Mobile Robots

TEXT AND REFERENCE BOOKS:

1. Henry C.Mishkoff, “Understanding Artifical Intelligence”
2. V S Janakiraman, “Foundation of Artificial Intelligence and Expert Systems”
3. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”

Artificial Intelligence Lab-List of Experiments

1. Write a program to implement DFS
2. Write a program to implement BFS
3. Write a Program to find the solution for travelling salesman Problem
4. Write a program to implement Simulated Annealing Algorithm
5. Write a program to find the solution for wampus world problem
6. Write a program to implement 8 puzzle problem
7. Write a program to implement Towers of Hanoi problem
8. Write a program to implement A* Algorithm
9. Write a program to implement Hill Climbing Algorithm
10. Build a bot which provides all the information related to you in college.