સૌરાષ્ટ્ર યુનિવર્સિટી

એકેડેમિક વિભાગ

યુનિવર્સિટી કેમ્પસ, યુનિવર્સિટી રોડ, રાજકોટ-૩૬૦૦૦૫

ફોન નં.(૦૨૮૧)૨૫૭૮૫૦૧ એક્સટે. નં.૨૦૨, ૩૦૪ ફેક્સ નં.(૦૨૮૧)૨૫૭૬૩૪૭ ઈ-મેઈલ : academic@sauuni.ac.in

નં.એકે/કોમ્પ્યુટર સાયન્સ/૨૬૧૯૬૫/૨૦૨૫

તા.૧૪/૧૦/૨૦૨૫

B.C.A., B.Sc.(IT) and B.Sc. CS

પરિપત્ર:-

સૌરાષ્ટ્ર યુનિવર્સિટીની કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા હેઠળની સ્નાતક કક્ષાના બી.એસસી.(કોમ્પ્યુટર સાયન્સ) ના અભ્યાસક્રમ ચલાવતી સર્વે સંલગ્ન કોલેજોના આચાર્યશ્રીઓને આથી જાણ કરવામાં આવે છે કે, તા.૦૪/૦૯/૨૦૨૫ના રોજ B.C.A., B.Sc.(IT) and B.Sc. CS ના પ્રેક્ટીકલ પેપર પ્રશ્નના નિવારણ માટે કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની એક સભાનું આયોજન કરેલ જેના કાર્યવાહી નોંધનાં ઠરાવ ક્રમાંક '૦૨' માં નીચે મુજબ ઠારાવવામાં આવેલ છે.

- B.C.A. અને B.Sc.(IT) ના અભ્યાસક્રમમાં SEE તરીકે જ્યાં પ્રેક્ટીકલ સબ્જેક્ટમાં થીયરી એક્ઝામ લેવાતી હતી તેની જગ્યાએ હવે ત્યાં પ્રેક્ટીકલ એક્ઝામ લેવાની રહેશે જેમાં ખાસ સેમેસ્ટર-1 થી સેમેસ્ટર-4 કોલેજ લેવલે અને સેમેસ્ટર-5 થી સેમેસ્ટર-6 માં યુનિવર્સિટી દ્વારા પ્રેક્ટીકલ એક્ઝામ લેવામાં આવશે
- જે વિદ્યાર્થીને પ્રેક્ટીકલ સબ્જેક્ટની થીયરી એક્ઝામમાં ATKT આવેલ હોય તેમને જૂની સ્કીમ પ્રમાણે જ થિયરીની એક્ઝામ આપવાની રહેશે જ્યાં સુધી ATKT રહેશે ત્યાં સુધી જુની સ્કીમ પ્રમાણે એમને પ્રેક્ટીકલ થિયરીની એકઝામ આપવાની રહેશે

ઉપરોક્ત મુદ્દા ક્રમાંક (૧) અને (૨) માટે માન. કુલપતિશ્રીને અધિકાર મંડળોની બહાલીની અપેક્ષાએ મંજુરી આપવા ભલામણ કરવામાં આવેલ છે.

- ઉપર્યુક્ત બાબત માટે B.C.A. અને B.Sc.(IT) સત્ર (૧) થી (૬) ના અભ્યાસક્રમો રજુ કરવામાં આવ્યા અને તેને માન. કુલપતિશ્રી સાહેબને અધિકાર મંડળોની બહાલીની અપેક્ષાએ મંજુરી આપવા ભલામણ કરવામાં આવેલ છે.
- ડીનશ્રી, તથા ચેરપર્સનશ્રી, કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની તેમજ કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની તા.૦૪/૦૯/૨૦૨૫ની કાર્યવાહી નોંધનાં ઠરાવ ક્રમાંક '૦૨' અન્વયે સત્ર ૧ થી ૬ અભ્યાસક્રમ અધિકાર મંડળોની બહાલીની અપેક્ષા મંજુરી આપવા માન.કુલપતિ સાહેબને ભલામણ કરેલ જે માન.કુલપતિશ્રીએ મંજુર કરેલ છે. જેથી સંબંધિત તમામે તે મુજબ તેની ચુસ્તપણે અમલવારી કરવી.

(મુસદ્દો કુલસચિવશ્રીએ મંજુર કરેલ છે.)

સહી/-
(ડૉ.આર.જી.પરમાર)
કુલસચિવ

બિડાણ:- ઉક્ત અભ્યાસક્રમ (સોફ્ટ કોપી)

રવાના કર્યું
14/10/25
એકેડેમિક ઓફીસર

પ્રતિ,

(૧) કોમ્પ્યુટર વિદ્યાશાખા હેઠળની B.C.A., B.Sc.(IT) and B.Sc. CS વિષય ચલાવતી સ્નાતક કક્ષાની સર્વે સંલગ્ન કોલેજોના આચાર્યશ્રીઓ તરફ
(૨) કોમ્પ્યુટર સાયન્સની વિષયની અભ્યાસ સમિતિના સર્વે સભ્યશ્રીઓ
(૩) ડીનશ્રી, કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા

નકલ જાણ અર્થે રવાના:-

૧. માન.કુલપતિશ્રી/કુલસચિવશ્રીના અંગત સચિવ

નકલ રવાના (યોગ્ય કાર્યવાહી અર્થે):-

૧. પરીક્ષા વિભાગ ૨. પી.જી.ટી.આર.વિભાગ ૩. જોડાણ વિભાગ

SAURASHTRA UNIVERSITY

RAJKOT – INDIA

CURRICULAM

of

4 Year UG Programme

Bachelor of Computer Application (Honours)

&

Bachelor of Computer Application
(Honours with Research)

(Semester - 3 and Semester – 4)

To be effective from June – 2024

B.C.A. (Honours) & B.C.A. (Honours with Research)
(Semester - 3 and Semester - 4)
Saurashtra University
To be effective from June – 2024

. Structure of Question Paper (50 Marks) for SEE

SEE (50 marks) Paper setting guide lines for all the semester

Question Paper contains 5 questions (each of 10 marks). Every question will be asked from respective unit as specified in the syllabus of each course. (i.e., Question-1 from Unit No.1 and remaining questions from their respective units). Every question is divided in three parts like (a), (b) and (c). Part (a) contains three objective type questions (not MCQ) like definition, reason, answer in one line, answer in one word etc., each of one mark and no internal option. Part (b) contains two questions each of two marks and student will attempt any one out of two. Part (c) contains two questions each of five marks and student will attempt any one out of two.

Saurashtra University **BCA Semester- 1/2/3/4/5/6**					
Time: 02:00					**Total marks: 50**
Q. 1	(a)	Attempt the following			03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following		Unit-1	02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 2	(a)	Attempt the following			03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following		Unit-2	02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 3	(a)	Attempt the following			03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following		Unit-3	02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 4	(a)	Attempt the following			03
		(1)			
		(2)			
		(3)		Unit-4	
	(b)	Attempt any one of the following			02
		(1)			
		(2)			

	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 5	(a)	Attempt the following		Unit-5	03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following			02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			

SEE (25 marks) Paper setting guide lines for all the semester

Question Paper contains 3 questions (Q.1 – 10 marks, Q.2 – 10 marks and Q.3- 05 marks). Q.1 is from unit-1, contains four questions each of five marks and student will attempt any two out of four. Q.2 is from unit-2, contains four questions each of five marks and student will attempt any two out of four. Q.3 is from unit-3, contains two questions each of five marks and student will attempt any one out of two.

Saurashtra University **BCA Semester- 1/2/3/4/5/6**				
Time: 01:00			**Total marks: 25**	
Q. 1	Attempt any two of the following		Unit-1	10
	(1)			
	(2)			
	(3)			
	(4)			
Q. 2	Attempt any two of the following		Unit-2	10
	(1)			
	(2)			
	(3)			
	(4)			
Q. 3	Attempt any one of the following		Unit-3	05
	(1)			
	(2)			

B.C.A. (Honours) & B.C.A. (Honours with Research)
(Semester - 3 and Semester - 4)
Saurashtra University
To be effective from June – 2024

B.C.A. SEM 3

Sr. No.	Type of Course	Course Title	Credit	CCE	SEE	Total
1	MAJOR	CS-15: C++ and Object Oriented Programming	4	50	50	100
2	MAJOR	CS-16: RDBMS Using Oracle	4	50	50	100
3	MAJOR	CS-17: Content Management System Using Wordpress	4	50	50	100
4	MDC	CS-18: Practical Based on CS -15, CS -16, CS -17	4	50	50	100
5	AEC	CS-19: Open Source Tools	2	25	25	50
6	SEC	CS-20: Network Technology and Administration	2	25	25	50
7	IKS	CS-21: Constitutional Values and Fundamental Duties	2	25	25	50
			22	**300**	**250**	**550**

B.C.A. SEM 4

Sr. No.	Type of Course	Course Title	Credit	CCE	SEE	Total
1	MAJOR	CS-22: Programming with Java	4	50	50	100
2	MAJOR	CS-23: Programming with C#	4	50	50	100
3	MAJOR	CS-24: Operating Systems Concepts with Unix/Linux	4	50	50	100
4	MINOR	CS-25: Practical Based on CS-22, CS – 23, CS-24	4	50	50	100
5	AEC	CS-26: Fundamentals of IoT	2	25	25	50
6	SEC	CS-27: Web Searching Technology and Optimization	2	25	25	50
7	VAC	CS-28: Digital Empowerment	2	25	25	50
			22	**300**	**250**	**550**

CCE = Continuous and Comprehensive Evaluation, SEE = Semester End Evaluation

B.C.A. (Semester – 3)

Sr. No.	Type of Course	Course Title	Credit
1	**MAJOR**	**CS-15:** C++ and Object-Oriented Programming	4
2	**MAJOR**	**CS-16:** RDBMS Using Oracle	4
3	**MAJOR**	**CS-17:** Content Management System Using Wordpress	4
4	**MDC**	**CS-18:** Practical Based on CS -15, CS -16, CS -17	4
5	**AEC**	**CS-19:** Open Source Tools	2
6	**SEC**	**CS-20:** Network Technology and Administration	2
7	**IKS**	**CS-21:** Constitutional Values and Fundamental Duties	2
		Total Credits of Semester 3	**22**

CS-15: C++ and Object Oriented Programming

Objectives:

- To provide OOP concepts, Input / Output data management, arrays in C++, functions, classes, objects, pointers and much more.
- Object-Oriented features, which allow the programmer to create objects within the code.

Prerequisites:

- Concepts of OOPs and their implementation.

Unit No.	Topic	Detail
1	**Principles of Object Oriented Programming Tokens, and Control Statements**	• Procedure – oriented programming • Object oriented programming paradigm • Basic concepts of object-oriented Programming • Benefits of object-oriented programming • Application of object-oriented programming • What is C++? • Application of C++ • Input/output operators • Structure of C++ program • Introduction of namespace • Tokens: keywords, identifiers, basic data types, user- defined types, derived data types, symbolic constants, type compatibility, declaration of variables, dynamic initialization of variables, reference variables • Operators in C++: ▪ scope resolution operator, ▪ member referencing operator, ▪ memory management operator, ▪ manipulators • Control structures ▪ Conditional control structure: simple if, if…else , nested if else, switch etc. ▪ Looping control structure: for, while , do…while
	Functions in C++	• The main function • Call by reference

		• Return by reference • Inline function • Default arguments • Const arguments • Functions overloading
2	**Classes and Objects, Constructor and Destructor**	• C structures revisited • Specifying a class • Local Classes • Nested Classes • Defining member functions, nesting of Member functions, private member function, making outside function inline • Arrays within a class • Memory allocation for objects • Static data member • Static member functions • Arrays of objects • Objects as function arguments • Friendly functions • Returning objects • Const member function • Pointer to members
		• Characteristics of constructor • Explicit constructor • Parameterized constructor • Multiple constructor in a class • Constructor with default argument • Copy constructor • Dynamic initialization of objects • Constructing two dimensional array • Dynamic constructor • MIL, Advantage of MIL • Destructors
3	**Operator Overloading and type conversion, Inheritance**	• Concept of operator overloading • Overloading unary and binary operators • Overloading of operators using friend Function • Manipulation of string using operators • Rules for operator overloading • Type conversions • Comparison of different method of conversion • Defining derived classes • Types of inheritance (Single, Multiple, Multi-level, Hierarchical, Hybrid) • Virtual base class & Abstract class

		• Constructors in derived class • Application of Constructor and Destructor in inheritance • Containership, Inheritance V/s Containership
4	**Pointer, Virtual Functions and Polymorphism, RTTI Console I/O Operations**	• Pointer to Object • Pointer to derived class • this Pointer • Rules for virtual function • Virtual function and pure virtual function • Run Time Type Identification (RTTI) • C++ Streams • C++ Stream Classes • Unformatted and formatted I/O operations • Use of Manipulators.

5	**Working with Files, Exception Handling, Introduction to Template STL**	• File Stream Classes
		• Opening and closing a file
		• Error Handling
		• File Modes
		• File Pointers
		• Sequential I/O operations
		• Updating a file (Random access)
		• Command Line Arguments
		• Overview of Exception Handling
		  ◦ Need for Exception Handling
		  ◦ various components of exception handling
		• Introduction to templates
		  ◦ Class templates and Function templates
		  ◦ Member function templates
		  ◦ Overloading of template function
		  ◦ Non-type Template argument
		• Introduction to STL
		  ◦ Overview of iterators, containers

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 60 + 15 = 75

Reference Books:

- Complete Reference C++ by Herbert Schildt McGraw Hill Publications
- Computer Science- A Structured approach using C++ by Forouzan, Gilburg, THOMSON
- Object Oriented Programming in C++ - E.Balagurusamy, BPB
- Object Oriented programming in C++ by Robert Lafore, Pearson Education
- Mastering C++ - Venugopal
- The C++ Programming Language by Bjarne Stroustrup, Pearson Education

- Object Oriented Programmin in C++ - Robaret Laphore
- Let us C++ - Yashvant Kanitkar, BPB

Course Outcomes:

- Understand the concept and underlying principles of Object-Oriented Programming.
- Understand implementation issues related to object-oriented techniques.
- Apply the techniques of object-oriented programming to solve real problems
- Analyze, apply and write programs that make appropriate use of object-oriented functionality such as classes, overloading and inheritance
- Implement the file handling techniques for back-end storage problems solutions

CS-16: RDBMS Using ORACLE

Objectives:

- To provide the basic concept, theory and practices in design and implementation of DBMS.
- To be able to handling different type of data transaction by using SQL commands.

Prerequisites:

- Theoretical as well as practical knowledge of database management system.

Unit No.	Topic	Detail
1	**DBMS Overview, SQL, SQL *PLUS**	• Introduction to DBMS and RDBMS • Dr. E. F. Codd Rules • Importance of E. R. Diagram in RDBMS • Normalization • Introduction to SQL • SQL Commands and Datatypes • Introduction to SQL *PLUS • SQL *PLUS formatting commands • Operator and Expression • SQL v/s SQL *PLUS
2	**Managing Tables and Data, Data Control and Transaction Control Command**	• Creating, Altering & Dropping tables • Data Manipulation Command like Insert, update, delete • Different type of constraints and applying of constraints • SELECT statement with WHERE, GROUP BY and HAVING, ORDER BY, DISTINCT, Special operators like IN, ANY, ALL, BETWEEN, EXISTS, LIKE • Join (Inner join ,outer join, self join) • subquery, minus, intersect, union • Built in functions Numeric Functions: abs, ceil, cos, decode, exp, floor, greatest, least, log, max, min, rem, round , sin, sqrt, tan, trunc Character Functions: chr, concat, initcap, lower, lpad, ltrim, replace, rpad, rtrim, substr, trim, upper Date Functions: add_months, last_day, next_day, months_between, round (date), sysdate, trunc (date), systimestamp, to_date, to_char Aggregate Functions: Sum, Count, AVG, MAX, MIN • Creating user & role • Grant, Revoke command • What is transaction? • Starting and Ending of Transaction Commit, Rollback,

		SavePoint
3	**Other Oracle Database Objects, Concurrency control using lock**	• View • Sequence • Synonyms • Database Links • Overview of Index and their types • Cluster • Snapshot • Locks, Overview of Locking Issues, Lock types
4	**Introduction to PL/SQL, Advanced PL/SQL**	• SQL v/s PL/SQL • PL/SQL Block structure • Language construct of PL/SQL (Variable, Basic and Composite Data Type, Conditions, Looping etc.) • %Type and %Rowtype • Using Cursor (Implicit, Explicit) • Exception Handling • Creating and Using Procedure • Package • Trigger • Creating Objects • Object in Database – Table • PL/SQL Tables, Nested Tables, Varrays
5	**Oracle Database Structure**	• Instance Architecture • Creating and Altering Database • Opening and shutdown Database • Initialization Parameter • Control Files, Redo Log Files • Concept of Tablespace • Rollback Segment • Import • Export • SQL *Loader

Seminar - 5 Lectures
Expert Talk - 5 Lectures (Managing a Multitenant Environment using Oracle 12c)
Test - 5 Lectures

Total Lectures 60 + 15 = 75

Reference Books:

- Oracle Database 12c The Complete Reference (Oracle Press) by Bob Bryla , Kevin Loney – Oracle Press
- Oracle Database 12c SQL – Jason Price – Oracle Press
- Oracle Database 12c PL/SQL Programming by McLaughlin – Oracle Press
- SQL, PL/SQL The programming - Lang.Of Oracle Ivan Bayross – BPB

Course outcomes:

- Describe the fundamentals of data design and relation database concepts
- Design entity-relationship diagrams to represent database application scenarios
- Develop relational database
- Apply normalization techniques on relational database
- Describe the knowledge of transaction processing and various concurrency problems
- Apply knowledge of SQL queries to perform various database related operations
- Develop various PL/SQL programs

CS-17: Content Management System using WordPress

Objectives:

- Learn how to create custom themes and pages
- Work with custom post types and taxonomies
- In detail knowledge of the WordPress CMS backend
- Working with widgets and widget areas
- Working in default cms functions and extending its core

Prerequisites:

- Basic knowledge of web development and CMS

Unit No.	Topic	Detail
1	**Introduction, Installation & Configuration**	• What is Content Management System (CMS)? - Introduction of Wordpress - Features of Wordpress • Advantages & Disadvantages of Wordpress - Installation of Wordpress. - Wordpress Directory & file structure. - Dashboard overview - How to add, edit and delete page, category, post, tag. - Add new media file (image, pdf, doc etc.) & attach to post or page. • Gutenberg Introduction - Gutenberg Blocks (Paragraph, Heading, Subheading, Quote, Image, Cover Image, Gallery, Video, Audio, Columns, Code, List, Button, Embeds) • User Roles and Capabilities. - Setting (General, writing, Reading, Discussion, Media, Permalinks) • Updating Wordpress - One-click Update - Manual Update • Database Structure
2	**Theme**	• What is Theme? • How to install & activate theme • Theme Customize Options (Site Identity, Menus, Widgets, HomePage Settings, Additional CSS)
3	**Widget**	• What is widget & widget areas? • Widget Management - Available Widgets (Archive, Calendar, Categories, Navigation Menu, Meta, Pages, Recent Comments, Recent Posts, RSS, Search, Tag Cloud, Text, Image Gallery, Video, Audio, Custom HTML) • Inactive Sidebar (not used)

		• Inactive Widgets
	Plugin	• What is plugin? • How to install and activate plugin • Useful plugins for website - SEO Yoast - Contact Form 7 - WooCommerce - WP Super Cache - Regenerate Thumbnails - Advanced Custom Fields - All-in-One WP Migration - Custom Post Type Widgets • Inactive Sidebar (not used) Inactive Widgets
4	**Theme Development**	• Anatomy of a Theme: header.php, footer.php and sidebar.php • Template Files (style.css, index.php, page.php, home.php, archive.php, single.php, comments.php, search.php, attachment.php, 404.php, category.php, tag.php, author.php, date.php) • The Loop (have_posts (), the_post()) • Template Tags ○ General tags (wp_head(), get_footer(), get_header(), get_sidebar(), get_search_form(), bloginfo(), wp_title(), single_post_title(), wp_footer(), comments_template(), add_theme_support(), get_template_directory_uri(), body_class()) ○ Author tags (the_author(), get_the_author(), the_author_link(), get_the_author_link(), the_author_meta(), the_author_posts()) ○ Category tags (category_description(), single_cat_title(), the_category()) ○ Link tags (the_permalink(), get_permalink(), home_url(), get_home_url(), site_url(), get_site_url()) ○ Post tags (the_content(), the_excerpt(), the_ID(), the_tags(), the_title(), get_the_title(), the_date(), get_the_date(), the_time(), next_post_link(), previous_post_link(), posts_nav_link(),

		post_class()) o Post Thumbnail tags (has_post_thumbnail(), get_post_thumbnail_id(), the_post_thumbnail(), get_the_post_thumbnail()) o Navigation Menu tags (wp_nav_menu()) o Conditional Tags (is_archive(), is_category(), is_front_page(), is_home(), is_page(), is_single(), is_search(), is_attachment(), is_active_sidebar()) • functions.php file
5	**Advanced Development**	• Advanced Functions - add_action(), add_filter(), add_shortcode(), do_shortcode(), register_nav_menu() • Custom Post Types - Register_post_type(), register_taxonomy(), Display custom post type & taxonomy • Widget Area - register_sidebar(), dynamic_sidebar()

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 60 + 15 = 75

Reference Books:

- Build Your Own Wordpress Website: An ultimate guide for small business owners paperback by Wordpress Genie
- Teach Yourself VISUALLY Wordpress paperback – by George Plumley 3rd Edition.
- Wordpress for Beginners: A visual step-by-step guide to Mastering Word press Paperback – by Dr. Andy Williams.
- Wordpress to Go: How to build a Wordpress website on your own domain, from scratch, Even if you are a complete beginner paperback – by Sarah Mcharry (Author)

Course outcomes:

- Work with and configure the cms backend
- Know when to use a custom post type or custom field
- Extend the Wordpress cms core to match requirements
- Create stunning dynamic themes

CS-18: Practical Based on CS -15, CS -16, CS -17	
Objectives: • To apply theoretical concepts through practical application. • To develop practical skills in various aspects of OOP, DBMS and framework usages like WordPress. **Prerequisites:** • Knowledge of OOP • Knowledge of DBMS • Knowledge of Basic Web Development	
CCE- Continuous and comprehensive Evaluation as follow • The continuous Comprehensive Evaluation (CCE) for each subject will be conducted by the teacher of that subject. The teacher will decide how the evaluation will be done. Usually CCE includes things like class participation, case studies and presentation, assignments, tutorials, small test (announced or surprised), quizzes and attendance or a mix of these. • Students must submit their work for internal evaluation on time to time. • Another part of CCE is the mid-term exam, which is compulsory for all students. This exam will be conducted internally by the college.	50 Marks
SEE – Semester End Examination as per the following • Practical Exam is conducted by college using approved examiners (3 Hours duration) • Students must prepare a practical notebook/book for the final practical examination. (The practical book serves as a record of all practical work, observations, procedures and results performed during the semester in lab. It is essential for evaluation during the final practical examination)	50 Marks

CS-18: Practical Based on CS -15, CS -16, CS -17	**Total Marks - 100**	
Topics	CCE	SEE
CS-15: Practically implementation of C++ and OOP which includes:Functions in C++, Classes, Inline Function, Friend Function, Special Member Function, Inline Function, Default Arguments, Return by Reference, Friend Function, Private Member Function, Constructor, Multiple Constructor in a class, MIL, Inheritance, Compile Time Polymorphism, Run Time Polymorphism, Exception Handling, Operator Overloading , Type Conversion, Virtual Base Class, Virtual Function, Pure Virtual Function, Manipulators	20	20
CS-16 Practically implementation of RDBMS Using Oracle which includes: DDL, DCL, DML and TCL Statements, Constraints, Joins, Subquery, Built-in Functions, View, Sequence, PL/SQL Block, Cursor, Trigger, Package, Nested Table	15	15

CS-17 Designing of Web site in CMS includes WordPress which includes: Post, Pages, Plugins, Theme Creation, Widgets, Working with Functions.php , Shortcode, Custom Post Types	15	15

Course Outcomes:

- Able to get knowledge about a comprehensive understanding of Object-Oriented Programming and their features.
- Able to gain practical experience in designing WordPress website including posts, pages, theme creation etc.
- Student will able to implement RDBMS features like procedure, trigger, view and other PL/SQL objects
- Student will get practical skills in OOP, WordPress website design, and implementation of RDBMS features using Oracle.

CS – 19 Open Source Tools

Objectives:

- Understanding Open Source Philosophy
- Identify and explore a range of open source tools.
- Understand the collaborative nature of open source development and the role of communities.
- Learn how to contribute to open source projects through code contrications, documentation, bug reporting etc.

Prerequisites:

- Basic Computer Skills
- Basic knowledge of Version Control

No	Topics	Details

1	**Open Source Softwares**	• Understanding Open Source Software
		  • Definition
		  • Principles
		  • History and evolution
		• Open-Source Licensing
		  • Overview
		  • Rights and responsibilities of users and developers under open source licenses
		  • Contracts & licenses and related issues
		• Application of Open sources
		• Open Sources Operating System:
		  • FEDORA
		  • UBUNTU

2	**Open Source Development and Collaboration**	• Version Control with Git
		  • Introduction to version control systems.
		  • Git fundamentals, repositories, commits, branches and merges
		• Open Source Project Management
		  • Overview of Project Management Methodologies (Agile)
		  • Tools for Project Planning, Task Tracking and Team Collaboration (Trello)
		  • Contributing to open source projects: Issue Tracking, Pull Requests, Code Reviews.

3	**Case Studies**	• Apache • Linux Operating System

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 30 + 15 = 45

Reference Books:

- "Producing Open Source Software: How to Run a Successful Free Software Project" by Karl Fogel
- "Git Pocket Guide: A Working Introduction" by Richard E. Silverman
- "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford
- KailashVadera, Bhavyesh Gandhi, “Open Source Technology”, Laxmi Publications Pvt. Ltd 2012, 1st Edition.
- Fadi P. Deek and James A. M. McHugh, “Open Source: Technology and Policy”, Cambridge Universities Press 2007.

Course Outcomes:

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

CS – 20 NETWORK TECHNOLOGY AND ADMINISTRATION

Objectives:

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize with the basic taxonomy and terminology of the computer networking area and advanced networking.
- Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.

Prerequisites:

- Basic knowledge of computer networking.

No	Topics	Details

1	**Basics of Network, Network Models and LAN Sharing**	• Network concepts
		  ○ What is network?
		  ○ Use of network
		• Network model: peer – to – peer, client – server
		• Network Services
		  ○ File service,
		  ○ Print service,
		  ○ Comm. service,
		  ○ Data base service,
		  ○ Security service,
		  ○ Application service
		• Network Access Methods
		  ○ CSMA / CD,
		  ○ CSMA / CA,
		  ○ Token passing,
		  ○ Polling
		• Network Topologies: Bus, Ring, Star, Mesh, Tree, Hybrid
		• Advanced Network Topologies Ethernet, CDDI, FDDI
		• Communication Methods
		  ○ Unicasting,
		  ○ Multicasting,
		  ○ Broadcasting
		• OSI reference model with 7 layers
		• TCP/IP network model with 4 layers

2	**Transmission Media Multiplexing & Switching Concepts Network devices**	• Transmission Media
		  ○ Types of Transmission media
		  ○ Guided media
		  ○ Co – Axial Cable, Twisted Pair Cable,
		  ○ Crimping of Twisted pair cable, Fiber Optic Cable
		• Unguided media
		  ○ Infrared, Laser, Radio, Microwave, Bluetooth tech.

		• Different Frequency Ranges
		• Multiplexing & De-multiplexing
		• Multiplexing Types
		  ○ FDM,
		  ○ TDM,
		  ○ CDM,
		  ○ WDM
		• Switching Tech.
		  ○ Circuit Switching,
		  ○ Message Switching,
		  ○ Packet Switching
		• CABLE NETWORK DEVICES
		• LAYER1 DEVICES
		  ○ LAN CARD,
		  ○ MODEM,
		  ○ DSL & ADSL
		  ○ HUB(Active, Passive, Smart hub), REPEATER
		• LAYER2 DEVICES
		  ○ SWITCH(Manageable, non- manageable)
		  ○ BRIDGE(Source route, Transactional)
		• LAYER3 DEVICES
		  ○ ROUTER,
		  ○ LAYER3 SWITCH
		  ○ BROUTER,
		  ○ GATEWAY,
		  ○ Network Printer
		• WIRELESS NETWORK DEVICES
		  ○ Wireless switch,
		  ○ Wireless router,
		• ACCESSPOINT

3	**Network Protocols and IP Addressing**	• Packets &Protocols
		• Conn. Oriented protocols –TCP & connection less Protocols - UDP
		• TCP/IP STACK, HTTP, FTP, SMTP, POP3, SNMP,
		• TELNET, ARP, RARP, IPX/SPX, AppleTalk,
		• NetBIOS Name PROTOCOL
		• L2CAP, RFCOMM Protocol
		• What is ip address?
		• Types of ip address
		• ipv4
		  ○ Class structure, subneting, super netting
		• ipv6
		  ○ Basic structure of ipv6

		○ Implementation of ipv6 • Migration from ipv4 to ipv6

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 30 + 15 = 45

Reference Books:

- Networking Essential - Glenn Berg Tech. Media
- MCSE Self-Paced Training Kit (Server 2003) Data Communication and Networking - B A Forouzan
- Networking Essential - Glenn Berg Tech. Media
- MCSE Self-Paced Training Kit (Server 2003)
- Data Communication and Networking - B A Forouzan

Course outcomes:

- Understand various types of computer networks
- Enumerate the layers of the OSI model and TCP/IP
- Understand principles of LAN design such as topology and configuration
- Apply transmission media and various networking devices to establish networks
- Compare and Analyze various spread spectrum and multiplexing techniques
- Understand network industry trends such as: Routing Protocols, IP Addresses, Error Detection

CS – 21 Constitutional Values and Fundamental Duties

Objectives:

- Enrich students with knowledge and relevance of the Constitution.
- Develop awareness about Duties and Values
- Inculcate a sense of Constitutionalism in thought and action.

Prerequisites:

- Fundamentals of constitution.

No	Topics	Details
1	**The Constitution of India – an Introduction**	• Federal Republic, Rule of Law, Separation of Powers • Sovereignty, Democracy • Secularism and Sarva Dharma Sama Bhava
2	**Fundamental Duties**	• Understanding Fundamental duties and their constitutional significance • Fundamental duties – Article 51A [(a) – (k)] • Analysis of UCCA • Overview of Article 370 and its implications
3	**Constitutional Values**	• Justice: Social, Political, Economic • Liberty: Thought, Expression, Belief, Faith, Worship • Equality: Equality before law & equal application of laws

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 30 + 15 = 45

Reference Books:

- Durga Das Basu, et al., introduction to the Constitution of India (LexisNexis, 26th edn, 2022().
- Mahendra Pal Singh, V. N. Shukla's Constitution of India, (Eastern Book Company, Laucknow, 13th revised edn. 2017)
- Leila Seth, We, the Children of India: The Preamble to Our Constitution (New Delhi, Puffin Books, Penguin Books India, 2010)

Course Outcomes:

- Understand the Constitution and its relevance.
- Appreciate the values and goals embedded in Constitution.
- Recognise the importance of fundamental duties enshrined in the Constitution.
- Apply the spirit of fundamental values and duties in everyday national life.

B.C.A. (Semester – 4)

Sr. No.	Type of Course	Course Title	Credit
1	**MAJOR**	**CS-22:** Programming with Java	4
2	**MAJOR**	**CS-23:** Programming with C#	4
3	**MAJOR**	**CS-24:** Operating Systems Concepts with Unix/Linux	4
4	**MINOR**	**CS-25**: Practical Based on CS-22, CS – 23, CS-24	4
5	**AEC**	**CS-26:** Fundamentals of IoT	2
6	**SEC**	**CS-27:** Web Searching Technology and Optimization	2
7	**VAC**	**CS-28:** Digital Empowerment	2
		Total Credits of Semester 4	**22**

CS-22: Programming with Java

Objectives:

- To provide fundamental concepts of Object-Oriented Programming and familiar with Java environment and its applications.
- To be able to understand control structures, classes, methods and argument passing and iteration graphical user interface basics programming and documentation style.

Prerequisites:

- Basic knowledge of object-oriented approach in programming with basic skills using Java.

Unit No.	Topic	Detail
1	**History, Introduction and Language Basics, Classes and Objects**	• History and Features of Java • Java Editions • JDK, JVM and JRE • JDK Tools • Compiling and Executing basic Java Program • Java IDE (NetBeans and Eclipse) • Data Type (Integer, Float, Character, Boolean) • Java Tokens: Keyword, Literal, Identifier, Whitespace, Separators, Comments, Operators: • Operators: Arithmetic, Relational, Boolean Logical, Bitwise Logical, Assignment, Unary, Shift, Special operators • Java Keywords (assert, strictfp, enum) • Type Casting - Decision Statements (if, switch) • Looping Statements (for, while, do..while) • Jumping Statements (break, continue, return) • Array (One Dim., Rectangular, Jagged) • Command Line Argument Array
		• OOP Concepts (Class, Object, Encapsulation, Inheritance, Polymorphism) • Creating and using Class with members • Constructor • finalize() method • Static and Non-Static Members • Overloading (Constructor & Method) • Varargs, IIB (Instance Initialization Block) in Java

2	**Inheritance, Java Packages**	• Universal Class (Object Class) • Access Specifiers (public, private, protected, default, private protected) • Constructors in inheritance • Method Overriding • Interface, Object Cloning, • Nested and Inner Class • Abstract and Final Class • Normal import and Static Import • Introduction to Java API Packages and imp. Classes o java.lang, o java.util o java.io, o java.net o java.awt, o java.awt.event o java.applet, o java.swing • java.lang Package Classes (Math, Wrapper Classes, String, StringBuffer) • java.util Package Classes (Random, Date, GregorianCalendar, StringTokenizer, Collection in Java • Vector, HashTable, LinkedList, SortedSet, Stack, Queue, Map • Creating and Using UserDefined package and sub-package
3	**Exception Handling, Threading and Streams (Input and Output)**	• Introduction to exception handling • try, catch, finally, throw, throws • Creating user defined Exception class - Thread and its Life Cycle (Thread States) • Thread Class and its methods • Synchronization in Multiple Threads (Multithreading) • Deamon Thread, Non-Deamon Thread • Stream and its types (Input, Output, Character, Byte) • File and RandomAccessFile Class • Reading and Writing through Character Stream Classes (FileReader, BufferedReader, FileWriter, BufferedWriter) • Reading and Writing through Byte Stream Classes (InputStream, FileInputStream, DataInputStream, OutputStream, FileOutputStream, DataOutputStream) • StreamTokenizer Class • Piped Streams, Bridge Classes: InputStreamReader and OutputStreamWriter • ObjectInputStream, ObjectOutputStream
4	**JavaFx Basics and**	• Basic Structure of JAVAFX program,

	Event-driven programming and animations	• Panes, • UI Control and Shapes, • Property binding, • the Color and the Font class, • the Image and Image-View class, • layout panes and shapes, • Events and Events sources, • Registering Handlers and Handling Events, • Inner Classes, anonymous inner class handlers, • mouse and key events, • listeners for observable objects, • animation
5	**JavaFx UI controls and multimedia**	• Labeled and Label • Button • Checkbox • Radiobutton • Textfield • Textarea • Combobox • Listview • Scrollbar • Slider • Video and Audio

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 60 + 15 = 75

Reference Books:

- Java: A Beginner's Guide – Jul 2014 by Herbert Schildt
- Java Programming (Oracle Press) by Poornachandra Sarang
- Java The Complete Reference, 8th Edition – by Herbert Schildt
- Ivor Horton's "Beginning Java 2" JDK 5 Edition, Wiley Computer Publishing.
- Ken Arnold, James Gosling, David Holmes, "The Java Programming Language", Addison-Wesley Pearson Education
- JavaFx A Beginners Guide by J. F. DiMarzio, McGraw Hill Computing
- Getting Started with JavaFx by Oracle: https://docs.oracle.com/javase/8/javafx/JFXST.pdf
- James Gosling, Bill Joy, Guy Steele, Gilad Bracha, "The Java Language Specifications", Addison-Wesley Pearson Education (3rd edition) Download at http://docs.oracle.com/javase/specs/

Course outcomes:

- Understand basic concepts and Java Programming Constructs
- Demonstrate Object Oriented Programming Concepts using JAVA
- Develop robust application by demonstrating professionally acceptable coding
- Design attractive user interface using AWT

CS-23: Programming with C#

Objectives:

- Demonstrate knowledge of object-oriented concepts design, user experience and functional requirements C# .Net Application.

Prerequisites:

- Basic knowledge of C# programming language and .Net environment.

Unit No.	Topic	Detail
1	**.NET Framework, and Visual Studio IDE, Language Basics**	• Introduction to .Net Framework • Features / Advantages • CLR, CTS and CLS • BCL / FCL / Namespaces • Assembly and MetaData • JIT and types • Managed Code and Unmanaged Code • Introduction to .NET Framework and IDE versions • Different components (windows) of IDE • Types of Projects in IDE: Console, Windows, Web, Setup, etc. • Data Types: Value Type & Reference Type • Boxing and UnBoxing • Operators: Arithmetic, Relational, Bitwise, etc. • Arrays: One Dimensional, Rectangular, Jagged • Decisions: If types and switch case • Loops: for, while, do..while, foreach
2	**Class and Inheritance, Property, Indexer, Pointers, Delegates, Event, Collections**	• Concept of Class, Object • Encapsulation, Inheritance, Polymorphism • Creating Class and Objects • Methods with “ref” and “out” parameters • Static and Non-Static Members • Constructors • Overloading Constructor, Method and Operator • Inheritance • Sealed Class & Abstract Class • Overriding Methods • Interface inheritance • Creating and using Property • Creating and using Indexer • Creating and using Pointers (unsafe concept) • Creating and using Delegates (Single / Multicasting) • Creating and using Events with Event Delegate • Collections: ArrayList, HashTable, Stack, Queue, SortedList

		and their differences.

3	**Windows Programming**	• Creating Windows Application
		• MessageBox class with all types of show() method
		• Basic Introduction to Form and properties
		• Concept of adding various Events with event parameters
		• Different Windows Controls
		  ○ Button,
		  ○ Label
		  ○ TextBox,
		  ○ RadioButton
		  ○ CheckBox,
		  ○ ComboBox
		  ○ ListBox,
		  ○ PictureBox
		  ○ ScrollBar,
		  ○ TreeView
		• Menu: MenuStrip, ContextMenuStrip, ToolStrip
		  ○ Timer
		  ○ Panel and GroupBox
		• Dialog Boxes
		  ○ ColorDialog,
		  ○ FontDialog,
		  ○ SaveFileDialog
		  ○ OpenFileDialog
		• MDI Concept with MDO Notepad
		• Concept of Inheriting Form

4	**Database Programming with ADO .NET**	• Concept of Connected and Disconnected Architecture
		• Data Providers in ADO.NET
		• Connection Object
		• Connected Architecture:
		  ○ Command,
		  ○ DataReader
		• Disconnected Architecture:
		  ○ DataAdapter,
		  ○ DataSet,
		  ○ DataTable, DataRow, DataColumn,
		• DataRelation, DataView Data Binding
		• GridView Programming

5	**User Controls (Components), Crystal Reports, Setup Project**	• Creating User Control with Property, Method, Event • Using User Control in Windows, • Creating Crystal Reports , Types of Reports • Report Sections • Formula, Special Field and Summary in Report • Types of Setup Projects • Creating Setup Project

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 60 + 15 = 75

Reference Books:

- Pro C# 5.0 and .NET 4.5 Framework (By: Andrew Troelse)
- Head First C# - (By: Jennifer Greene, Andrew Stellman)
- C# 5.0 Unleased – (By: Bart De Smet)
- Adaptive Code Via C# - (By: Gary McLean Hall)
- C# .NET Programming Black Book – Steven Holzner – Dreamtech Publications
- Introduction to .NET Framework – Wrox Publication
- Microsoft ADO .NET – Rebecca M. Riordan, Microsoft Press

Course outcomes:

- Use the Microsoft Visual Studio development environment to create a windows application
- Understand the basics of object-oriented programming, CLR and .NET framework
- Demonstrate C# programming constructs to solve given problem
- Perform CRUD operations in windows application
- Use the trace and debug utility that are provided with Visual Studio .NET
- Develop, configure and deploy windows application

CS-24: Operating Systems Concepts with Unix / Linux

Objectives:

- To provide the basic feature, function and interface with the hardware and application software to run the computer smoothly.

Prerequisites:

- Basic knowledge of operating system and it's functionality along with its types

Unit No.	Topic	Detail
1	**Introduction, Process and Thread, Process Scheduling**	• Meaning of OS • Functions of OS • Features of OS • OS Types (User Point of View) • OS Types (Features Point of View)

		• Process Definition
		• Process States
		• Process State Transitions
		• Process Control Block
		• Context Switching
		• Threads
		  ○ Concept of multithreads
		  ○ Benefits of threads
		  ○ Types of threads

		• Types of Schedulers • CPU Scheduling Algorithms • FCFS • SJN • Round Robin • Priority Base Non-Preemptive • Priority Base Preemptive
2	**Deadlocks, Memory Management**	• Deadlocks: Definition • Deadlock Prevention • Deadlock Avoidance • Deadlock Detection • Physical Memory and Virtual Memory • Memory Allocation • Internal and External fragmentation • Contiguous Memory Allocation • Noncontiguous Memory Allocation • Virtual Memory Using Paging • Virtual Memory Using Segmentation

3	**Getting Started with Unix, Unix Shell Command**	• Unix Architecture
		• Unix Features
		• Types Of Shell (C, Bourn, Korn)
		• Unix File System
		• Types Of Files
		  ○ Ordinary Files
		  ○ Directory Files
		  ○ Device Files
		• Unix File & Directory Permissions

		• Connecting Unix Shell : Telnet • Login Commands passwd, logout, who, who am i, clear,uname • File / Directory Related Command ls, cat, cd, pwd, mv, cp, ln, rm, rmdir, mkdir, chmod, chown, chgrp, find, more, less, head, tail, wc, touch, stat, alias, type • Operators in Redirection & Piping <, >, <<, >>, | • Finding Patterns in Files grep, fgrep, egrep • Working with columns and fields cut, paste, join • Tools for sorting :sort, uniq • Comparing files : cmp, comm, diff • Changing Information in Files: tr, sed • Examining File Contents : od • Tools for mathematical calculations: bc, factor • Monitoring Input and Output :tee, script • Tools For Displaying Date and Time: cal, date • Communications : telnet, ping • Process Related Commands: ps, sleep
4	**Text Editing with vi and nano Editor, Shell Programming**	• Introduction of vi editor • Modes in vi • Switching mode in vi • Cursor movement • Screen control commands • Entering text, cut, copy, paste in vi editor • Introduction of nano editor
		• Shell Keywords • Shell Variables • System variables PS2, PATH, HOME,LOGNAME, MAIL, IFS, SHELL, TERM, MAILCHECK

		• User variables set, unset and echo command with shell variables • Positional Parameters • Interactive shell script using read and echo • Decision Statements o if then fi o if then else fi o if then elif else fi o case esac • test command • Logical Operators • Looping statements o for loop o while loop o until loop o break, continue command • Array • Function • Various shell script examples
5	**Getting Started with Linux, Linux Booting, Linux Admin (Ubuntu)**	• History of Linux • GNU, GPL Concept • Open Source & Freeware • Structure and Features of Linux • Installation and Configuration of Linux o Using with Ubuntu • Startup, Shutdown and boot loaders of Linux
		• Linux Booting Process o LILO Configuration o GRUB Configuration
		• Creating Linux User Account and Password • Installing and Managing Samba Server • Installing and Managing Apache Server • Configure Ubuntu's Built-In Firewall • Working with WINE

Seminar	-	5 Lectures
Expert Talk	-	5 Lectures
Test	-	5 Lectures

Total Lectures 60 + 15 = 75

Reference Books:

- Operating System Concept, Abraham Silberschatz, Peter B. Galvineg Gagne, Wiley-Indian Edition, 9th Edition

- Operating Systems, Internals and Design Principles, William Stallings, Seventh Edition
- Unix Shell Programming – Y. Kanetkar – Bpb Publications
- Unix Concepts and Applications – Sumitabha Das
- The complete reference Linux, Richard Petersen, McGraw Hill, Sixth Edition

Course outcomes:

- Understand design and implementation aspects of modern operating system
- Acquire knowledge of four major OS components: process management, memory management, file systems, and input/output mechanisms
- Analyze and compare various process scheduling algorithms
- Learn the concepts, design, and structure of the UNIX operating system
- Design shell scripts using various UNIX utilities

Hands-On (Not to be asked in the examination):

- Installation of Unix / Linux
- User and Group Creation
- Demo of Various Applications available in Unix / Linux like Star Office, Games and other productivity tools
- Demo of GNOME, KDE Desktops in Linux

CS-25: Practical Based on CS-22, CS – 23, CS-24	
Objectives: • To apply theoretical concepts through practical applications. • To develop practical skills in various aspects of JAVA, C# .Net and Shell Scripting **Prerequisites:** • Knowledge of Object Oriented Programming • Knowledge of DBMS • Knowledge of computer operating	
CCE- Continuous and comprehensive Evaluation as follow • The continuous Comprehensive Evaluation (CCE) for each subject will be conducted by the teacher of that subject. The teacher will decide how the evaluation will be done. Usually CCE includes things like class participation, case studies and presentation, assignments, tutorials, small test (announced or surprised), quizzes and attendance or a mix of these. • Students must submit their work for internal evaluation on time to time. • Another part of CCE is the mid-term exam, which is compulsory for all students. This exam will be conducted internally by the college.	50 Marks
SEE – Semester End Examination as per the following • Practical Exam is conducted by college using approved examiners (3 Hours duration) • Students must prepare a practical notebook/book for the final practical examination. (The practical book serves as a record of all practical work, observations, procedures and results performed during the semester in lab. It is essential for evaluation during the final practical examination)	50 Marks

CS-25: Practical Based on CS-22, CS – 23, CS-24	**Total Marks - 100**	
Topics	CCE	SEE
CS-22 Practically implementation of Java Program which includes: Java Array, Command Line Argument Array, OOP concepts, Java Access Specifiers and Inheritances, IIB and VARARGs in JAVA, Java Packages, Exception Handling , Threading , Classes of JavaFx, JavaFx UI Controls, JavaFx Listeners, Event Handlers, JavaFx Multimedia	20	20
CS-23 **Practically implementation of C# program which includes:** Jagged Array, Keywords in C# (Ref, out), Indexers, Delegates, Collections, Windows Controls, Dialog Controls, MDI Form, Connected Architecture, Disconnected Architecture	20	20

CS-24 Unix Shell script which includes: Redirection and piping, File and directory related command, Finding Pattern in Files, Positional Parameters, Decision, Looping Statements in Script, Logical Operators in Script	10	10

Course Outcomes:

- Able to get knowledge about a comprehensive understanding of Object Oriented Programming and their features.
- Student will get practical skills in GUI Development, clear the basics of OS, and implementation of shell scripting too.

CS – 26 FUNDAMENTALS OF IoT

Objectives:

- Understand the fundamental concepts and principles of the Internet of Things.
- Explore the architecture, components and technologies used in IoT systems.
- Learn about different communication protocols and standards for IoT.
- Gain insights into the design considerations and challenges in developing IoT solutions.
- Acquire practical skills in designing and implementing IoT systems.

Prerequisites:

- Basic knowledge of computer networks and protocols
- Familiarity with programming languages such as C / C++

Unit No.	Topic	Detail
1	**Introduction to IoT**	• Introduction to the Internet of Things (IoT) • History and Evolution of IoT • Key Concepts and Definitions • Applications and Use Cases of IoT • Challenges and Opportunities in IoT
2	**IoT Architecture and Technologies**	• Conceptual Framework • IoT Architecture Overview • Technology behind IoT • Sources of the IoT • M2M Communication • IoT Examples

3	**Hardware for IoT**	• Sensors
		• Digital Sensors
		• Actuators
		• Radio Frequency Identification (RFID) Technology
		• Wireless sensor networks
		• Overview of IoT supported Hardware platforms:
		  ○ Arduino
		  ○ Netduino

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 30 + 15 = 45

Reference Books:

- "Internet of Things (A Hands-on Approach)" b Arshdeep Bahga and Vijay Madisetti
- "Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz
- "Designing Connected Products: UX for Consumer Internet of Things" by Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light, and Alfred Lui

Course Outcomes:

- Explain the concept and significance of the Internet of Things in various domains.
- Describe the architecture and components of IoT systems, including sensors, actuators, and communication protocols.
- Analyze different IoT communication protocols and select appropriate protocols for specific IoT applications.
- Identify design considerations and challenges in developing scalable and secure IoT solutions.

CS – 27 WEB SEARCHING TECHNOLOGY AND OPTIMIZATION

Objectives:

1. Understand basic of search engines and reflecting
2. Understand SEO objectives and defining site audience.
3. Apply and Implement SEO friendly website with all SEO concept.
4. Understand keyword research and apply it for website developments.
5. Understand the new trends of digital technologies.

Prerequisites:

Basic knowledge of SEO, search engine and E-commerce.

No	Topics	Details
1	**Search Engine Basics and Understanding SEO Objectives**	• The Mission of Search Engines & Market Share • Human Goals of Searching & Determining Searcher Intent • How People Search & How Search Engines Drive Commerce • Eye Tracking & Click Tracking: Natural vs. Paid • Understanding Search Engine Results & Algorithm-Based Ranking Systems • Determining SEO Objectives & Setting Goals • Understanding Audience & Finding Niche • Major Elements of Planning & Identifying Competitors
2	**Implementing SEO-friendly Website**	• Making Site Accessible to Search Engines • Creating Optimal Information Architecture • Root Domains, Subdomains, and Microsites • Optimization of Domain Names/URLs & Keyword Targeting • Content Optimization & Duplicate Content Issues • Controlling Content with Cookies and Session IDs • Content Delivery and Search Spider Control • Redirects & Content Management System (CMS) Issues • Optimizing Flash & Best Practices for Multilanguage/Country Targeting
3	**Keyword Research and Tracking Results**	• Theory Behind Keyword Research & Traditional Approaches • Site Content Analysis & Keyword Research Tools • Determining Keyword Value & Leveraging the Long Tail • Opportunities in Vertical Search & Optimizing for Different Types • Tracking Results & Measuring Success • Measuring Search Traffic & Tying SEO to Conversion and ROI • Competitive and Diagnostic Search Metrics • Performance indicators for Long Tail SEO & Future Trends in SEO

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 30 + 15 = 45

Course outcomes:

- Understand the main elements that help a website rank organically and in the paid search space in Google.
- Learn how to perform keyword research using Google's free tools.
- Learn how to develop landing pages that are search engine friendly.
- Learn how to carry out inbound linking practices.

Reference Books:

- The Art of SEO : Mastering Search Engine Optimization By Eric Enge, Stephan Spencer, Rand
- Fishkin, Jessie C Stricchiola, O'Reilly Media, 3rd Edition October, 2015
- Google SEO Bible, Beginner's Guide to SEO, ISBN-978-1700098733, moaml mohmmed, 2019
- SEO Warrior: Essential Techniques for Increasing Web Visibility By John I Jerkovic, O'Reilly Media, November, 2009

CS – 28 Digital Empowerment

Objectives:

- Understand the digital world and need for digital empowerment
- Create awareness about Digital India
- Explore, communicate and collaborate in cyberspace
- Building awareness on cyber safety and security

Prerequisites:

- Basic computer literacy and familiarity with Operating System..

Unit No.	Topic	Detail

1	**Digital Inclusion and Digital Empowerment**	• Needs and Challenges
		• Vision of Digital India:
		  ○ DigiLocker
		  ○ E-Hospitals
		  ○ E-Pathshala
		  ○ SHIM
		  ○ E-Kranti (Electronic Delivery of Services)
		  ○ e-Health Campaigns
		• Public utility portals of Govt. of India such as RTI, Health, Finance, Income Tax filing, Education

2	**Communication and Collaboration in the Cyberspace**	• Electronic Communication: electronic mail, blogs, social media • Collaborative Digital platforms • Tools / Platforms for online learning • Collaboration using file sharing, messaging, video conferencing
3	**Towards Safe and Secure Cyberspace**	• Online security and privacy • Threats in the digital world: Data breach and Cyber Attacks • Blockchain technology • Security Initiatives by the Govt. of India

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures

Total Lectures 30 + 15 = 45

Suggested Books, References and Online Resources

- David Sutton, “Cyber Security: A Practitioner’s guide”, BCS Learning & Development Limited, UK, 2017
- https://www.mha.gov.in/document/downloads/cyber-safety-handbook
- Rodney Jones and Christoph Hafner – “Understanding digital Literacies: A practical Introduction”, Routledge Books, 2nd Edition, 2021.

- https://www.digitalindia.gov.in
- https://www.digilocker.gov.in
- https://www.cybersafeindia.in
- https://www.meity.gov.in/cyber-suraskshit-bharat-programme

Course Outcomes:

- Use digital services in daily life.
- Develop skills to communicate and collaborate in cyberspace using social platforms,
  teaching / learning tools.
- Understand the significance of security and privacy in the digital world.
- Evaluate ethical issues in cyber world.

	BCA-3 **CS-15: C++ and Object Oriented Programming** **Minimum following exercise should be performed by the students during the semester**
1.	Write a C++ code to print "Hello, World!" to the console.
2.	Write a C++ code to take two integers as input and print their sum.
3.	Write a C++ code to swap two numbers using a temporary variable.
4.	Write a C++ code to find the maximum of two numbers using if-else.
5.	Write a C++ code to check whether a number is even or odd.
6.	Write a C++ code to create a simple calculator using switch-case.
7.	Write a C++ code to check whether a given year is a leap year.
8.	Write a C++ code to print numbers from 1 to N using a loop.
9.	Write a C++ code to compute the factorial of a number using a loop.
10.	Write a C++ code to print the Fibonacci series up to N terms.
11.	Write a C++ code to reverse the digits of a given number.
12.	Write a C++ code to check whether a number is prime or not.
13.	Write a C++ code to print the multiplication table of a number.
14.	Write a C++ code to compute the sum of digits of a number.
15.	Write a C++ code to check whether a number or string is a palindrome.
16.	Write a C++ code to check whether a number is an Armstrong number.
17.	Write a C++ code to convert a binary number to a decimal number.
18.	Write a C++ code to convert a decimal number to binary.
19.	Write a C++ code to count the number of digits in an integer.
20.	Write a C++ code to compute the power of a number using a loop.
21.	Write a C++ code to perform linear search in an array.
22.	Write a C++ code to perform binary search in a sorted array.
23.	Write a C++ code to find the maximum element in an array.
24.	Write a C++ code to find the sum of elements in an array.
25.	Write a C++ code to reverse an array in-place.
26.	Write a C++ code to count the frequency of each element in an array.
27.	Write a C++ code to sort an array using bubble sort.
28.	Write a C++ code to merge and sort two arrays.
29.	Write a C++ code to find duplicate elements in an array.
30.	Write a C++ code to find the length of a string without using inbuilt functions.
31.	Write a C++ code to compute factorial using recursion.
32.	Write a C++ code to print Fibonacci series using recursion.
33.	Write a C++ code to compute GCD of two numbers using recursion.
34.	Write a C++ code to compute LCM of two numbers.
35.	Write a C++ code to find the sum of numbers from 1 to N using recursion.
36.	Write a C++ code to check if a number is prime using recursion.
37.	Write a C++ code to solve the Tower of Hanoi problem.
38.	Write a C++ code to check if a string is a palindrome using recursion.
39.	Write a C++ code to implement a power function using recursion.
40.	Write a C++ code to swap two numbers using pointers and functions.
41.	Write a C++ code to demonstrate the use of pointers.
42.	Write a C++ code to swap two variables using pointers.

43.	Write a C++ code to access array elements using pointers.
44.	Write a C++ code to create and use a dynamic array.
45.	Write a C++ code to demonstrate pointer to pointer (double pointer).
46.	Write a C++ code to return a pointer from a function.
47.	Write a C++ code to demonstrate pointer arithmetic.
48.	Write a C++ code to demonstrate a dangling pointer issue.
49.	Write a C++ code to define a class and create objects.
50.	Write a C++ code to demonstrate the use of constructors and destructors.
51.	Write a C++ code to define a constructor with default values.
52.	Write a C++ code to implement a copy constructor.
53.	Write a C++ code to overload the + operator for a class.
54.	Write a C++ code to demonstrate function overloading.
55.	Write a C++ code to use static variables and functions in a class.
56.	Write a C++ code to access private members using a friend function.
57.	Write a C++ code to implement single inheritance.
58.	Write a C++ code to implement multilevel inheritance.
59.	Write a C++ code to implement multiple inheritance.
60.	Write a C++ code to use virtual functions for runtime polymorphism.
61.	Write a C++ code to implement an abstract class using pure virtual functions.
62.	Write a C++ code to override a base class function in a derived class.
63.	Write a C++ code to demonstrate the use of virtual destructor.
64.	Write a C++ code to use function templates.
65.	Write a C++ code to create a class template.
66.	Write a C++ code to specialize a template for a specific data type.
67.	Write a C++ code to handle exceptions using `try`, `catch`, and `throw`.
68.	Write a C++ code to define and throw a custom exception class.
69.	Write a C++ code to write text to a file.
70.	Write a C++ code to read text from a file.
71.	Write a C++ code to copy contents from one file to another.
72.	Write a C++ code to count the number of words in a file.
73.	Write a C++ code to read a file line by line.
74.	Write a C++ code to append content to a file.
75.	Write a C++ code to write and read binary data using a file.
76.	Write a C++ code to check if a file exists before opening.
77.	Write a C++ code to read a CSV file and process its data.
78.	Write a C++ code to manage student records using file operations.
79.	Write a C++ code to use a parameterized constructor to initialize class data.
80.	Write a C++ code to overload constructors in a class.
81.	Write a C++ code to create a constructor that initializes data using member initializer list.
82.	Write a C++ code to demonstrate constructor overloading in a class with default arguments.
83.	Write a C++ code to implement a class with a private constructor and a static method (Singleton pattern).
84.	Write a C++ code to initialize an object using a copy constructor.

85.	Write a C++ code to show constructor call order in inheritance.
86.	Write a C++ code to demonstrate constructor in a derived class calling base class constructor.
87.	Write a C++ code to demonstrate single inheritance with public access specifier.
88.	Write a C++ code to use private inheritance and show what gets inherited.
89.	Write a C++ code to demonstrate multilevel inheritance.
90.	Write a C++ code to implement multiple inheritance and resolve name conflicts.
91.	Write a C++ code to show constructor call sequence in multiple inheritance.
92.	Write a C++ code to use virtual base classes to solve diamond problem.
93.	Write a C++ code to override a function in derived class and call base class version.
94.	Write a C++ code to demonstrate protected inheritance.
95.	Write a C++ code to use `std::list` to perform insertions and deletions at both ends.
96.	Write a C++ code to demonstrate the use of `std::deque`.
97.	Write a C++ code to iterate over a `std::map` using iterator and range-based for loop.
98.	Write a C++ code to demonstrate a `std::set` with custom comparator.
99.	Write a C++ code to use `std::pair` and `std::tuple` for returning multiple values.
100.	Write a C++ code to remove duplicates from a vector using `std::sort` and`std::unique`.

BCA- 3
RDBMS Using Oracle
Minimum following exercise should be performed by the students during the semester

(1) Generate following tables and perform the queries:

Employee:

Emp_id	Emp_name	Post	Mgr_id	Hire_date	Salary	ommission	Dept_id
12345	BansiPadia	Clerk		18-11-1990	15000		10
13254	Radha Sharma	alesman	12345	25-11-1990	12000	4000	20
12675	TanmayDomadiya	resident		05-07-1985	124000		10
13457	Suraj Prakash	alesman	12345	12-08-1992	14650	1700	30
14567	Seema Lava	ogrammer	13654	17-07-1994	35000		10
16754	Krupa Vala	ogrammer	13654	13-08-1992	30000		10
14562	Kalpana Joshi	Clerk		17-09-1990	17000		40
12435	KevalRaval	alesman	12345	20-10-1990	18000	6500	20
13654	ParthVasoya	Analyst	12675	27-12-1995	90000		10
18654	Ramesh Vasoya	Analyst	12675	15-05-1993	85000		10

Department

Dept_no	Depart_Name	Location
10	IT	Anand
20	Sales	Bombay
30	Purchase	Surat
40	Finance	Anand
50	Marketing	Surat

a. Display all the details of salesman from Employee table.
b. List out all the salesman whose name starts with 'K' letter.
c. Find out the employee whose designation is clerk.
d. List out all the employees whose salary is between 20000 to 90000.
e. Display all the details of employee who are working in department 10.
f. Display the employees whose salary is greater than 25000 and department no is 10.
g. List out name, designation and salary of all the employees.
h. Calculate the total salary of an employees.
i. List out all the details of department, where location is Anand.
j. Display the information of department whose department no is either 10,20 or 40.
k. Display the information of department no. 50.
l. Display the name of the department either IT, Sales or Marketing.
m. Display all the details of employee who are either an Analyst or salary greater than 50000.
n. Display all the employees who are not a president.
o. List out department name and location from department table.
p. Display distinct Salary from employee table.
q. Retrieve all the details of an employee who were join during the 1990s.

(2) Create a following table report should have title, related column formatting and display sum of item price for all the orders.

Table: Order_detail - Order_no, Order_date, Item, Item_price, Discount

(3)	Create a following table and solve the given query.
	Table: Emp - Emp_no, Emp_name, company, salary, city
	I. Find the name of al employee who work for an InfoSystem Ltd.
	II. Find all the employee details who don't work for a InfoSystem Ltd. (Using Subquery)
	III. Find all the employee who works in city "Mumbai".
(4)	Create a following table and solve the given query.
	Table :Item_Master - Item_id, Item_name, price, company_code
	I. Write a SQL query to find all the item with a price between Rs. 200 and Rs. 600.
	II. Write a SQL query to calculate the average price of all items of manufactures which company code is 16.
	III. Write a SQL query to find the name and price of the lowest item.
(5)	Create a following table and solve the given query.
	Table: Client - C_no, c_name, city, state, post, salary
	a. List out all the client who are Managers.
	b. Display the details of all clients who are located in Gujarat state and Salary is less than10000.
	c. List out al client who are either Clerk or Manager.
	d. Display all the clients who are working as a Developer.
	e. Display all the clients who are located in Surat and name start with 'A' character.
(6)	Create a following table and solve the given query.
	Table : Student - Rollno, name, city, address, DOB, age, class, mobileno
	i. Insert 5 records in the table.
	ii. Display the records of students whose age is less than or equal to 18.
	iii. Display all students name in Ascending order and Age in Descending order.
	iv. Update the student city to "Jamnagar" for all records whose city is "Rajkot".
	v. Update the class to Sem-3 for all records whose rollno is 10.
	vi. Delete all students' records whose age is greater than 20.
	vii. Delete the student record whose city is Jamnagar and Class is not Sem-3.
(7)	Create following table and perform the queries:
	Table: Salesman- Sid , sname, city, sal, comm.
	a. List out all the name of salesman who is situated in Mumbai.
	b. Find out maximum commission taken by salesman.
	c. List out the name of salesman who has a salary equal to 3000.
	d. Change the city of salesman to Puna whose id is 8.
	e. Delete the entire salesman whose salaries are equal to 3500.
	f. Find out all salesman whose names starts with 's ' character.
	g. Display all salesman who are situated in Bombay, Delhi or Puna.
(8)	Create a following table and solve the given query.
	Table: Production - P_no, description, profit, unit, qty, s_price, c_price
	I. List the various products available from the production table.
	II. Change the cost price of Pen to Rs. 500.
	III. Delete all the products where the quantity is equal to 100.
	IV. List out products whose selling price is greater than 500 and less than or equal to 750.

(9)	Create a following tables and solve the given query.
	Table :Emp - Empno (primary key), ename (NOT NULL), job, mgrno (NOT NULL), hiredate, salary, comm, deptno (Foreign key)
	Table: Department - Deptno (Primary key), dname (NOT NULL), location
	i. Display all employee whose name start with 'S' or 'R'.
	ii. Display job title (Don't repeat) present in the company.
	iii. List all the employee having salary more than 'Amar' and less than 'Karan'.
(10)	Create a following tables and solve the given query.
	Table: Deposit - Ac_no (Primary key), cname, bname (Foreign key), amount, adate
	Table: Branch - Bname (Primary key), city
	Table: Customer - Cname (Primary key), city
	Table: Borrow - Loanno, cname (Foreign key), bname (Foreign key), amount
	i. Give the name of customers having living in Mumbai and Branch city Nagpur.
	ii. Give the name of cutomers having the same living city as their Branch city.
	iii. Give the name of customers who are borrowers as well as depositors and having living city Nagpur.
	iv. List name of borrowers having deposit amount greater than 1000 and loan amount greater than 2000.
	v. Display loan no, loan amount, account no and deposit amount of customers living in city Nagpur.
	vi. Display branch city and living city of Anil.
(11)	Create a following tables and solve the given query.
	Table :Student_Master - Sno (Primary key), sname, c_code (Foreign key), city, age
	Table :Fee_Master - Sno (Foreign key), fee
	Table :Course_master - C_code (Primary key), c_name
	i. Display all the students whose fees is not pay.
	ii. List out all the students whose city either Jamnagar, Rajkot or Surat.
	iii. Display the students' course wise.
	iv. Display all students name, course name and students age who having age greater than 23 and less than 27 and doing 'BCA'.
(12)	Create tables as per following and solve the queries.
	Table: Emp_mast - Emp_no(pk), emp_name, dept_code (fk), desi
	Table: Dept_mast - Dept_no(pk), dept_name
	Table: Salary_mast - Emp_no(fk), sal, hra, ma, pf
	i. Create a query to display all the employee name, department name and total salary i.esal + hra + ma – pf, who are having salary more than 2000 and less than 5000.
	ii. Create a update query to fill hra = 100 if a employee is having salary more than 2000 else fill hra = 200.
	iii. Delete all the employee information from emp_mast who are having department "Computer".
(13)	Consider the following tables and solve the given queries:
	Salesman:salesman_id(Primary key), name, city, commission
	Order:ord_id (Primary key), purchase_amt, order_date, cust_id(Foreign Key),
	salesman_id(Foreign key)
	Customer:cust_id(Primary Key), cname, city, grade, salesman_id(Foreign key)
	Emp_department:dept_no(Primary key), deptname

	Emp_details: Emp_id(Primary key), emp_fname, emp_lname, salary,
	dept_no(Foreign key)
a.	Write a query to find those customers with their name and those salesmen with their name and city that lives in the same city. (*table*: salesman, customer)
b.	Write a SQL statement to find the names of all customers along with the salesmen who works for them. (table: customer, salesman)
c.	Write a SQL statement to display all those orders by the customers not located in the same cities where their salesmen live. (Table: Salesman, customer, order)
d.	Write a SQL statement that finds out each order number followed by the name of the customers who made the order. (Table: orders, customer)
e.	Write a query that produces all customers with their name, city, salesman and commission, who served by a salesman and the salesman works at a rate of the commission within 12% to 14%. (salesman, customer)
f.	Write a SQL statement that produces all orders with the order number, customer name, commission rate and earned commission amount for those customers who carry their grade is 200 or more and served by an existing salesman. (Table: salesman, customer, orders)
g.	Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000. (order, customer)
h.	Write a SQL statement to know which salesman is working for which customer. (customer, salesman)
i.	Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.
j.	Write a query in SQL to display all the data of employees including their department. (emp_department, emp_details)
k.	Write a query in SQL to find the names of departments where more than two employees are working. (emp_department, emp_details)
(14)	Create table as per following and solve the queries.
	Table: Orders - Ord_no, purch_amt, ord_date, customer_id, salesman_id
i.	Write a query to display the columns in a specific order like order date, salesman id, order number and purchase amount from for all the orders.
ii.	Write a query which will retrieve the value of salesman id of all salesman getting orders from the customers in orders table without any repeats.
iii.	Write a SQL query to display the order number followed by order date and the purchase amount for each order which will be delivered by the salesman who is holding the ID 5001.
iv.	Write a query to display the orders according to the order number arranged by ascending order.
(15)	Create tables as per following and solve the queries.
	Table: Customer - C_id, cname, city, grade, salesman_id
	Table: Salesman - Sid, name, city, commission
i.	Write a query to display all the information for those customers with a grade of 200.
ii.	Write a query to display all customers with a grade above 100.
iii.	Write a query to display all customers in New York who have a grade value above 100.
iv.	Write a query to display all customers, who are either belongs to the city New York or had a grade above 100.
v.	Write a query to display customer name, city and grade in such a manner that the customer holding highest grade will come first.
vi.	Write a query to find out those customer name and salesman name who are

	situated within the same city.
(16)	Create following tables and perform the given queries.
Table: Salesman -	Sid, name, city, commission
Table: Customer -	C_id, cname, city, grade, salesman_id
I.	Write a SQL statement to find the names of all customers along with the salesman who works for them.
II.	Write a query to display all salesman and customer located in London. (UNION)
III.	Write a query to display salesman and their cities.
(17)	Create table and perform the given queries.
Table: emp_details -	Emp_no, fname, lname, emp_dept
i.	Write a query to find the last name of all employees without duplicate.
ii.	Write a query to display all the data of employees that work in the department 57.
iii.	Write a query to find the data of employees whose last name is Doshi or Joshi.
(18)	Create a following table and solve the given query.
Table :Item_Master -	Item_id, Item_name, price, company_code
Table: Company_master -	Com_id, com_name
i.	Write a query to display all the data from the item_master including each item's producer company.
ii.	Write a query to display the item name, price and company name of all the products.
iii.	Write a query to display the average price of items of each company showing the name of the company.
iv.	Write a query to display the names of the company whose products have an average price larger than or equal to Rs. 350.
(19)	Generate a PL/SQL block to find Odd or Even number.
(20)	Generate a Pl/SQL block to enter any number and print whole table of the entered number.
(21)	Write a PL/SQL program to find entered year is Leap year or not.
(22)	Generate a PL/SQL block to print 1 to 5 number using while loop.
(23)	Generate a PL/SQL block to print 1 to 10 number using simple loop.
(24)	Generate a PL/SQL block to print 1 to 10 number using for loop.
(25)	Generate a PL/SQL block to enter rollno, name, mark1, mark2, mark3 and find out the total, percentage and appropriate class.
(26)	Generate a PL/SQL block to print 1^2, 3^2, 5^2….. using while loop.
(27)	Generate a PL/SQL block to print the sum of 1, 4, 7 ….n value.
(28)	Generate a PL/SQL block to print the 0 1 1 2 3 5 8 ….n series.
(29)	Generate a PL/SQL block to enter any number and print its factorial value.
(30)	Write a PL/SQL program to check whether a given number is positive, negative or zero.
(31)	Write a PL/SQL program to check whether a given character is letter or digit.
(32)	Generate a PL/SQL block to print the prime numbers between 1 to 50.
(33)	Generate a PL/SQL block to display all the details of an employee, whose ID is 45.
(34)	Write a PL/SQL block that inserts a new employee into the employees table only if they are not already present.
(35)	Create a following table and write down PL/SQL procedure to display records using explicit cursor.Table : Student - Id (primary key), fname, lname, city
(36)	Demonstrate the use of explicit cursor with ISOPEN, NOTFOUND attributes.

(37)	Generate a Cursor to select the five highest paid employees from the employee table.
(38)	Write a PL/SQL block that uses an explicit cursor to fetch employee names and salaries.
(39)	Write a PL/SQL block that attempts to divide two numbers and handles division by zero exceptions.
(40)	Write a PL/SQL block that retrieves all the available employees in the table.
(41)	Write a recursive procedure that prints numbers in descending order from a given number.
(42)	Write a procedure to delete records from the employees table based on a given department ID.
(43)	Write a PL/SQL block that checks if a specific employee exists in the employees table.
(44)	Write PL/SQL block for displaying sid, sname, DOB, address, city from student table using procedure with call block.
(45)	Write a PL/SQL code for stored procedure to change student name in capital letter by passing Roll no and raise appropriate error if the student is not exist in student table.
(46)	Create a procedure that counts the number of employees in a particular department.
(47)	Create a procedure to add record into table student.
	Table: Student - Sid, sname, age, class
(48)	Create a package that contain two procedure to add record and to delete record. Choose a suitable table.
(49)	Write a PL/SQL code to find out square of number using INOUT parameter of procedure.
(50)	Write a PL/SQL code to find out minimum number using IN and OUT parameter of Procedure.
(51)	Write a PL/SQL code for trigger that perform delete operation on Main table and deleted record should be inserted into Backup table.
(52)	Write a PL/SQL code that perform insert and update record with less than 5000 salary then it raise error using trigger.
(53)	Write a database trigger which will not allow any transaction after office hours (after 7:00 PM) and on Sunday.
(54)	Write a trigger to record any changes made to the salary column in the salary_history table.
(55)	Write an update trigger on client master table. The system keep records are being updated. The old values of updated should be added in the Temp table.
	Table: Temp - C_no, name, operation, o_date, uid
(56)	Write a code in PL/SQL to create a trigger that prevents updates on a certain column during specific hours of the day.
(57)	Create a trigger that automatically records any insertions into the audit_log table whenever a new employee is added.
(58)	Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found.
(59)	Write a PL/SQL code to create a package that includes a procedure to calculate the factorial of a number and a function to check if a number is prime.
(60)	Create a package to handle employee salary operations: one procedure to increase salary, one to decrease salary, and one to fetch salary details.

BCA-3 **CS-17 Content management system using Word press** **Minimum following exercise should be performed by the students during the semester**	
(1)	Install XAMPP/WAMP and set up WordPress locally.
(2)	Install WordPress on live hosting (demo with free hosting).
(3)	Explore WordPress dashboard & create a short summary.
(4)	Change site title, tagline, and upload favicon/logo.
(5)	Configure permalink settings.
(6)	Set default category for posts.
(7)	Set time zone, language, and reading settings.
(8)	Install & activate a free theme.
(9)	Install & activate a premium theme (demo).
(10)	Customize theme colors, fonts, and background.
(11)	Create custom homepage & blog page.
(12)	Create a child theme and activate it.
(13)	Use theme editor to modify CSS.
(14)	Add and manage widgets in different positions.
(15)	Create a custom menu with dropdowns and external links.
(16)	Create Home, About, Services, and Contact pages.
(17)	Create multiple blog posts with categories & tags.
(18)	Insert images, videos, and audio into posts.
(19)	Add featured image to posts and pages.
(20)	Create password-protected post.
(21)	Create sticky posts.
(22)	Schedule a post and update it.
(23)	Create custom post formats (quote, gallery, video).
(24)	Create a page template and assign it.
(25)	Install and configure Contact Form 7.
(26)	Install and configure SEO plugin (Yoast).
(27)	Install caching plugin & check site speed.
(28)	Create multiple users with roles (Admin, Editor, Author, Subscriber).
(29)	Manage comments (approve, spam, delete).
(30)	Disable comments for specific posts/pages.
(31)	Configure media settings (thumbnail, medium, large).
(32)	Organize media library with folders plugin.
(33)	Install and configure WooCommerce plugin.
(34)	Create products with categories and variations.
(35)	Add coupon codes in WooCommerce.
(36)	Backup site and restore from backup.
(37)	Export and import WordPress site data.

BCA-4 **CS-22: Programming with Java** **Minimum following exercise should be performed by the students during the semester**	
(1)	Write a program to print "Hello World".
(2)	Write a program to calculate simple Interest.
(3)	Write a program to convert the amount from US$ to Indian Rupee.
(4)	Write a program to print first 10 natural numbers: 1 2 3 4 5 …….. 10
(5)	Write a program to print: 1 101 1001 10001
(6)	Write a program to print: 1 12 123 1234 1234
(7)	Write a program to find out whether the given number is odd or even.
(8)	Write a program to print first 100 odd numbers: 1 3 5 7 9 …….. 199
(9)	Write a program to find out maximum number out of given 3 numbers.
(10)	Write a program to find out max1, max2, max3 among 10 values.
(11)	Write a program to print 10 given numbers in ascending order.
(12)	Write a program to check whether a given string is palindrome or not.
(13)	Write a program to check whether the number is positive or negative or zero.
(14)	Write a program to find the maximum of an array using recursion.
(15)	Write a program to find the sum of an array using recursion.
(16)	Write a program to calculate the sum of all elements in a jagged array of integers.
(17)	Write a program to accept an array of strings in the command line and arrange them in alphabetical order.
(18)	Write a program to accept an integer using command line and print whether a given number is prime or not.

(19)	Write a program to find the area of circle and rectangle using command line argument.
(20)	Define a Room class with attributes length, breadth and area. Define a suitable constructor and method to display the details of Room.
(21)	Write a program to define a Car class with three attributes: brand (String), model (String), and year (int). Include a constructor, setter methods, and a method to display complete information.
(22)	Define a class GarbageDemo that overrides the finalize() method to perform cleanup activities before object destruction. Demonstrate calling finalize().
(23)	Write a program to create a class Counter with a static variable count and a non-static variable number. Increment count in constructor and display both number and count.
(24)	Write a program to create a class Calculator demonstrating method overloading with different add methods (two ints, three ints, two doubles).
(25)	Write a program to create a class Student demonstrating constructor overloading (no args, name only, name + age).
(26)	Write a program to create a class SumCalculator demonstrating variable-length arguments (varargs) to calculate sum.
(27)	Write a program to create a class Demo demonstrating Instance Initialization Block (IIB) execution before constructor.
(28)	Write a program to create a class Person demonstrating public, private, protected, and default access specifires. Show accessibility from same and different packages.
(29)	Write a program to create parent class Animal with method eat(), child class Dog with method bark(), demonstrating single inheritance.
(30)	Write a program to create a multilevel inheritance hierarchy: Animal → Mammal → Dog. Demonstrate method calls from all levels.
(31)	Write a program to create parent class Animal, child classes Dog and Cat. Demonstrate hierarchical inheritance with methods eat(), bark(), and meow().
(32)	Write a program to create parent class Animal with method sound(), child class Dog overriding sound(). Demonstrate method overriding.
(33)	Write a program to create parent class Shape with area() method. Child classes Circle and Rectangle override area() to calculate and display actual area.
(34)	Write a program to create interface Vehicle with start() method. Classes Car and Bike implement Vehicle with their own start() definitions.
(35)	Write a program to create interface Shape with area() method. Classes Circle and Rectangle implement Shape with their own area() calculations.

(36)	Write a program to create class Outer with inner class Inner. Demonstrate inner class instantiation and method calls.
(37)	Write a program to demonstrate abstract class Shape with abstract method area(). Subclass Circle implements area().
(38)	Write a program to demonstrate a final class MathConstants with constant PI. Show that it cannot be extended.
(39)	Write a program to demonstrate Math class: max, min, sqrt, abs, pow, rounding. Display results.
(40)	Write a program to demonstrate wrapper classes (Integer, Double, Character). Show autoboxing, unboxing, and parsing from strings.
(41)	Write a program to demonstrate String class operations: concatenation, length, comparison, substring check, uppercase/lowercase.
(42)	Write a program to demonstrate StringBuffer: append, insert, reverse, delete. Display results.
(43)	Write a program to demonstrate Random class: generate integers, floats, booleans and use them in calculations or events.
(44)	Write a program to demonstrate Date class: display current date/time and extract day, month, year. Format output.
(45)	Write a program to demonstrate StringTokenizer: split a sentence into words, count tokens, and display each token.
(46)	Write a program to create a Vector of integers: add, remove, and display elements. Show dynamic resizing.
(47)	Write a program to create a Hashtable for student roll numbers and names. Add, remove, and display key-value pairs.
(48)	Write a program to create a LinkedList of strings: add, insert, remove, and display elements using a loop.
(49)	Write a program to create a SortedSet of integers using TreeSet. Add elements and display in ascending order; show duplicates not allowed.
(50)	Write a program to create a Stack of integers: push, pop, and display elements. Demonstrate LIFO behavior.
(51)	Write a program to create a Queue of integers using LinkedList: add, remove, and display elements. Demonstrate FIFO behavior.
(52)	Write a program to create a HashMap of student IDs and marks: add, update, remove, and display key-value pairs.

(53)	Write a program to create a package mypackage with class Calculator. Create sub-package advanced with ScientificCalculator. Demonstrate usage.
(54)	Write a program to demonstrate try-catch: accept two numbers, perform division, handle ArithmeticException if denominator is zero.
(55)	Write a program to demonstrate finally block: perform operation in try-catch and include finally to print message.
(56)	Write a program to demonstrate throw keyword: throw exception for negative input in square root calculation, handle with catch.
(57)	Write a program to demonstrate throws keyword: method divide() throws ArithmeticException for zero denominator; handle in main.
(58)	Write a program to create a user-defined exception MarkOutOfBoundsException. Handle marks >100 or <0 in Student class.
(59)	Write a program to create a thread by extending the Thread class.
(60)	Write a program to create a thread by implementing the Runnable interface.
(61)	Write a program to demonstrate Thread life cycle.
(62)	Write a program to demonstrate thread priorities (MIN_PRIORITY, NORM_PRIORITY, MAX_PRIORITY).
(63)	Write a program to demonstrate synchronization when multiple threads access the same counter.
(64)	Write a program to demonstrate inter-thread communication using wait() and notify().
(65)	Write a program to create multiple threads and print numbers alternatively using them.
(66)	Write a program to demonstrate the use of join() method in threads.
(67)	Write a program to read a text file and count the number of words, lines, and characters.
(68)	Write a program to copy contents of one file into another using character streams.
(69)	Write a program to append user input into a file using FileWriter.
(70)	Write a program to demonstrate BufferedReader and BufferedWriter for file operations.
(71)	Write a program to demonstrate DataInputStream and DataOutputStream for reading and writing data.
(72)	Write a program to demonstrate PrintWriter for writing formatted data into a file.

(73)	Write a program to list all files and directories of a given path.
(74)	Write a program to demonstrate FileReader and FileWriter with exception handling.
(75)	Write a program to demonstrate basic structure of a JavaFX application: extend Application, override start(), display simple window.
(76)	Write a program to demonstrate different layout panes in JavaFX: BorderPane, HBox, VBox, with buttons, labels, or text fields.
(77)	Write a program to create a JavaFX window with UI controls like Button, Label, TextField, CheckBox, and shapes like Circle, Rectangle, Line.
(78)	Write a program to demonstrate property binding in JavaFX: two TextFields where one updates automatically when the other changes.
(79)	Write a program to demonstrate Color and Font classes: change color and font style of text in label or text node.
(80)	Write a program to display an image using Image and ImageView classes. Resize or move the image in the scene.
(81)	Write a program to demonstrate event handling in JavaFX: register handlers using inner classes, anonymous classes, or lambdas.
(82)	Write a program to handle mouse and key events in JavaFX. Detect clicks on shapes or key presses and display messages.
(83)	Write a program to attach listeners to observable objects like TextField or Slider. Update UI elements based on value changes.
(84)	Write a program to demonstrate simple animation in JavaFX: move a shape with TranslateTransition or fade-in/fade-out with FadeTransition.
(85)	Write a program to create a JavaFX window and display multiple Labels with different texts, colors, and fonts.
(86)	Write a program to create buttons in JavaFX and register event handlers to display messages when clicked. Demonstrate multiple buttons with different actions.
(87)	Write a program to create multiple CheckBox controls. Display which checkboxes are selected when a button is clicked.
(88)	Write a program to create a group of RadioButton controls. Ensure only one can be selected at a time and display selected option.
(89)	Write a program to accept user input using a TextField and display it in a TextArea when a button is clicked.
(90)	Create a JavaFX application with three ComboBoxes: Country, State, and City. Update

	state and city ComboBoxes based on user selection.
(91)	Write a program to create a ListView with multiple items. Allow user to select one or more items and display selected items.
(92)	Write a program to create a Scrollbar in JavaFX. Display its current value and allow user to change it.
(93)	Write a program to create a Slider control. Display its current value in real-time as the slider is moved.
(94)	Write a program to play a video or audio file using JavaFX Media and MediaPlayer classes with basic controls: play, pause, stop.
(95)	Create a registration form using JavaFX.

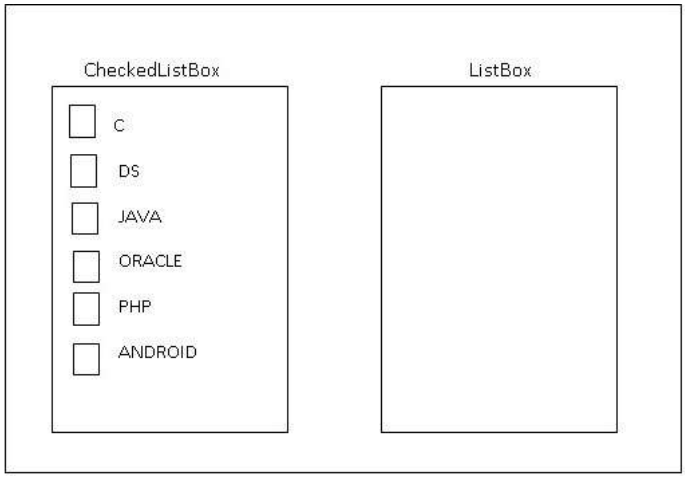
BCA SEM- 4

CS-23 Programming with C#

Minimum following exercise should be performed by the students during the semester

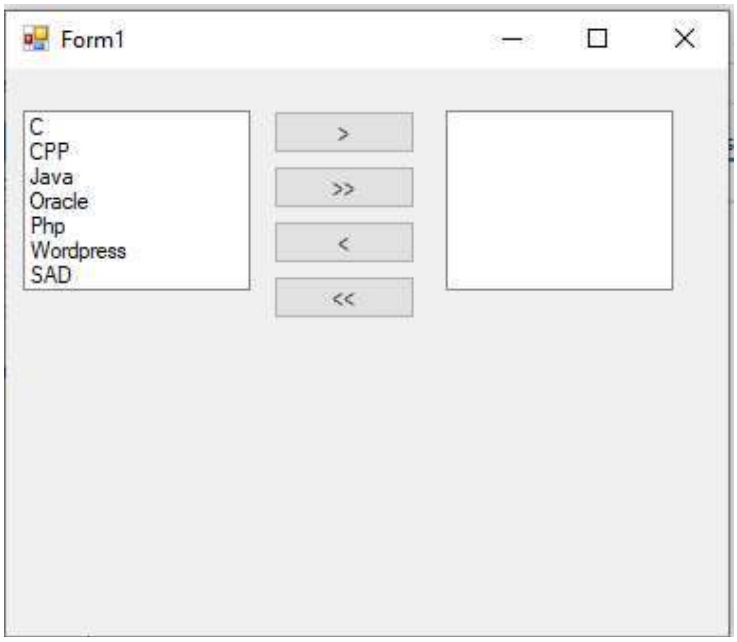
(1) Write a program to swap two numbers using arithmetic operators.
(2) Find the sum, difference, product, and quotient of two numbers.
(3) Use relational operators to compare two numbers and display the result.
(4) Use if-else to check if a number is positive, negative, or zero.
(5) Use nested if statements to find the largest among three numbers.
(6) Implement switch-case to print day of the week based on number input.
(7) Write a program using for loop to print first 10 natural numbers.
(8) Write a program using while loop to print even numbers up to 50.
(9) Write a program using do-while loop to accept numbers until zero is entered.
(10) Write a C# program to generate following series:
    a. 1 2 3…….n
    b. 1 3 5……..n
    c. 2+4+6…..n
    d. 1+4+7…….n
    e. 0 1 1 2 3 5……..n
    f. 1 2 2 4 8……..n
(11) Find the factorial of a number using loops.
(12) Calculate the sum of digits of a given number.
(13) Create a program to check if a number is palindrome.
(14) Implement a program to find prime numbers between 1 and 100.
(15) Use foreach loop to iterate over an array of strings.
(16) Create and display elements of a one-dimensional array.
(17) Create and display elements of a two-dimensional rectangular array.
(18) Create and display elements of a jagged array.
(19) Write a program to count vowels and consonants in a string.
(20) Create a program to reverse a string using loops.
(21) Write a program to merge two arrays.
(22) Write a program to remove duplicates from an array.
(23) Write a program to find the second largest element in an array.
(24) Create a jagged array of student marks and calculate average.
(25) Sort an array using bubble sort.
(26) Sort an array using selection sort.
(27) Use conditional operator (? :) to find maximum of two numbers.
(28) Implement nested switch-case to handle multi-level menu options.
(29) Write a program to simulate a simple calculator using loops and switch.
(30) Write a program to check leap year using if conditions.
(31) Write a program that uses break and continue statements inside loops.
(32) Create a class Student with properties and methods and generate details of the 3 students and display it.

(33) Implement encapsulation by making class members private and using properties.
(34) Create constructors (default and parameterized) in a class.
(35) Demonstrate constructor overloading.
(36) Create a class with method overloading.
(37) Implement operator overloading for + operator in a ComplexNumber class.
(38) Create a base class Person and derive Student and Teacher classes.
(39) Demonstrate method overriding using virtual and override keywords.
(40) Create an abstract class Shape and derive Circle and Rectangle classes.
(41) Create multicast delegate and combine multiple methods.
(42) Create event handlers and trigger events on property change.
(43) Create a class hierarchy with multiple levels of inheritance.
(44) Demonstrate polymorphism by calling overridden methods.
(45) Use List<T> collection and perform add, remove, search, and sort.
(46) Create a Windows Forms app with a Button and display a MessageBox on click.
(47) Create a form with TextBox to accept user input and display it on a Label.
(48) Create a simple calculator using buttons and TextBoxes.
(49) Create a form that changes background color on Button click.
(50) Create a ComboBox and populate it with country names.
(51) Create a ListBox and add/remove items dynamically.
(52) Use PictureBox to display an image.
(53) Implement MenuStrip with File and Help menus.
(54) Use ToolStrip with buttons and dropdowns.
(55) Create a form with Timer to update current time in a Label.
(56) Create a form with Panel and GroupBox to organize controls.
(57) Use ColorDialog to select color and change form background.
(58) Use FontDialog to change font of a TextBox.
(59) Use OpenFileDialog to select and display image in PictureBox.
(60) Use SaveFileDialog to save text from TextBox to file.
(61) Create a multiple-document interface (MDI) parent and child forms.
(62) Create a form with TreeView to display folder structure.
(63) Create a form with DataGridView to display tabular data.
(64) Validate form inputs before submission.
(65) Generate a Login Form with Username and password (password masking), and a submit button. When user will enter the value and press on submit button, then Message will be display on the screen 'Login Successful!'.
(66) Use tooltip to show help on controls.
(67) Develop a form that includes a **CheckedListBox** and a **ListBox**. When a user selects an option from the CheckedListBox, the corresponding item should be added to the ListBox. Similarly, when a user deselects an option from the CheckedListBox, the associated item should be removed from the ListBox.

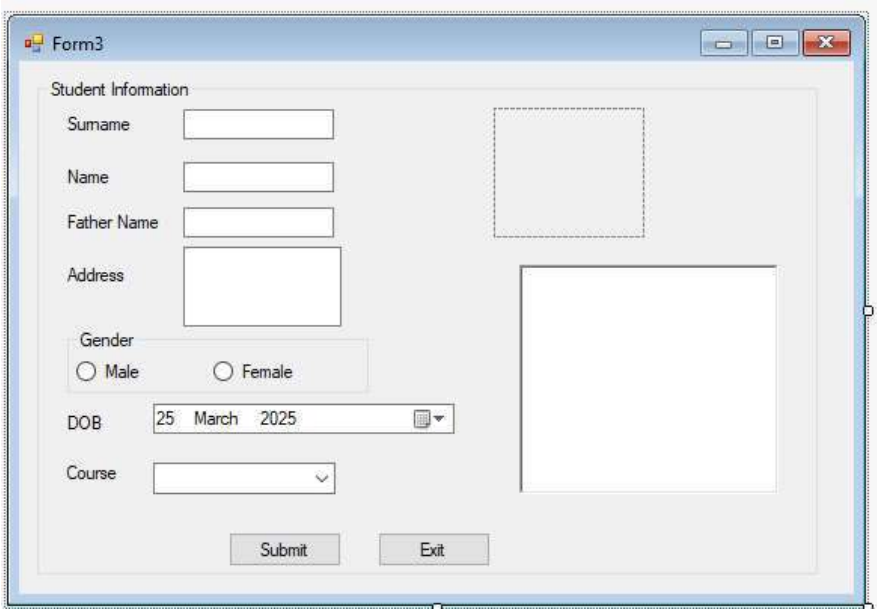


(68) Generate C# program for checkbox, when user will click on checkbox, the button is enabled, otherwise button is disabled

(69) Design a C# form with 2 different listboxes and move the items from one listbox to another based on the requirement either single or multiple items at a time.



(70) Generate a C# form for student Registration and entered data should be displayed into the RichTextBox.



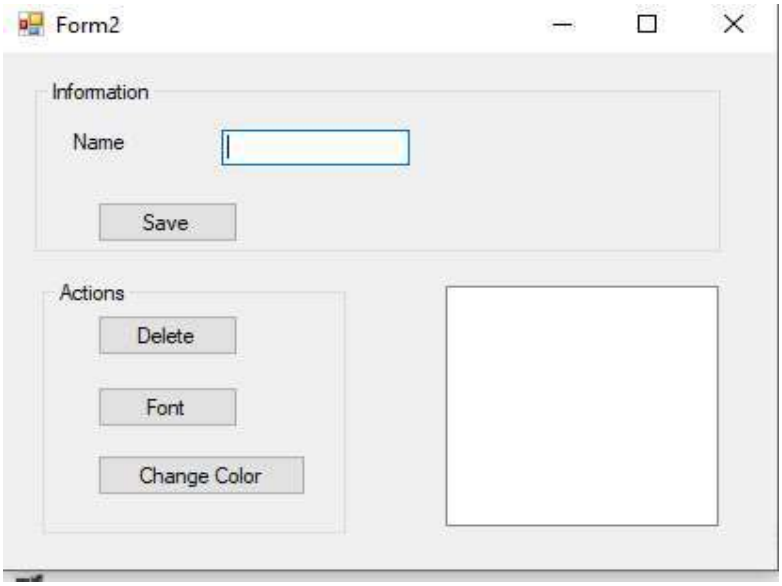
(71) Design a form with 3 buttons- Start, Calculate and Exit and 2 checkboxes- Sum and Average. When user will click on Start button, program should read numbers one by one. When 0 is entered, reading of numbers will be over. When calculate button is clicked, then, depending upon the choices of checkboxes, sum and/or average should be displayed.

(72) Generate a C# form to accept Name, City and Age through the textbox and when user will click on the submit button, welcome message will be displayed on the screen.

(73) Design a C# form with 1 checkedlistbox, 1- textbox and 1 – botton. When user will enter any item in the textbox and click on the button (Add Item), then entered item should be added in the given checkedlistbox.

(74) Create a C# form to demonstrate scroll bar for RGB function to change a colour of particular label/ form.

(75) Generate a C# form to insert any name in the textbox and it should be displayed into the listbox, and user can also delete the items/names from the listbox using delete button, change the font style/size/type using the font button and user can also change the color of listbox when click on the change color button.



(76) Develop a c# code for select records from the database using DataReader and also display it in the DataGridView .

(77) Create a form that allows adding new customers, updating their details, and deleting them from the database. Use **Connected Architecture** with SqlCommand andSqlDataReader for reading data.

(78) Generate a C# form application to perform CRUD operations (Connected/Disconnected) on Employee table.

(79) Generate a C# form application to navigate the Records.

(80) Implement disconnected data access by modifying DataSet and updating database.

(81) Generate a C# form to create master-detail relation using DataRelation.

(82) Generate a C# form to perform CRUD operation using ListBox.

(83) Generate a C# form to move data from one form to another form.

(84) Create a custom UserControl with properties and events.

(85) Add User control into a Windows Form.

(86) Create a MDI form with MenuBar and Dialogs.

(87) Create a Crystal Report based on database table and display it in a form.

(88) Add a `ColorDialog` to allow the user to select a color and change the background color of the form.

(89) Write a code for performing search operation on the Emp table which has fields empno, ename and salary. Display only those employees whose salary is less than 3000.

	BCA-4 **CS-24 Operating system concepts with Unix/Linux** **Minimum following exercise should be performed by the students during the semester**
(1)	Basic Linux Commands Learn and execute simple file & directory commands — pwd, ls, cd, mkdir, rmdir, rm, cp, mv, cat, touch. (Folder create, move, copy, delete, show path, etc.)
(2)	File Permissions Change file access rights and ownership using chmod, chown, chgrp. (Read, Write, Execute permission set karva.)
(3)	Viewing & Filtering Files Display and count file content using head, tail, wc, sort, uniq. (First/last lines, sorting, removing duplicates.)
(4)	Searching & Pattern Matching Search specific text or patterns using grep, egrep, fgrep.
(5)	Cut, Paste & Translate Extract specific columns, merge files and replace characters using cut, paste, tr.
(6)	Use of tee Command Show output on screen and save to a file at the same time using pipes and tee.
(7)	File Compression & Archiving Create and extract compressed archives using tar, gzip, gunzip, zip, unzip.
(8)	Process Management Check and control running processes using ps, top, kill, jobs, bg, fg, nice, renice.
(9)	Text Editing with VI/Nano Create and edit files, insert, delete, search, save and quit using VI or Nano editors.
(10)	Basic Shell Script Write a simple script using variables, echo, and read.
(11)	Decision Making Scripts Use conditions in shell scripts — if, if-else, case statements.
(12)	Looping in Shell Scripts Repeat tasks using for, while, and until loops.
(13)	Command Line Arguments Write shell scripts that accept and use command line arguments ($1, $2…).
(14)	Functions in Shell Scripts Create and use user-defined functions for reusable code.
(15)	To perform demonstration of WINE.
(16)	User & Group Management Create, delete, and manage users & groups — useradd, passwd, who, w, id.
(17)	Job Scheduling Schedule and automate tasks using at, batch, cron, crontab.
(18)	Searching Files in System Locate files and commands using find.
(19)	Backup & Restore Create and restore backups using tar and gzip.