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)

(As per NEP 2020)

To be effective from June – 2023

Ordinances, Regulations:

Ordinances:

O. B.C.A. – 1: Candidate for admission to the Bachelor of Computer Application must have passed standard 12th or equivalent examination from Gujarat Higher Secondary Board or any other board.

O. B.C.A. – 2: Candidate who have passed an equivalent examination from any other board or examining body and is seeking admission to the B.C.A. course will be required to provide necessary eligibility certificate.

O. B.C.A. – 3:

Definitions of Keywords:

1. **Academic Year:** An Academic Year is divided into two semesters and a semester of minimum 15 weeks comprises 90 working days.

2. **Programme:** An educational programme leading to award of the Certificate in Computer Application, Diploma in Computer Application, Bachelor of Computer Application, Bachelor of Computer Application (Honours) or Bachelor of Computer Application (Honours with Research).

3. **Course:** Usually referred to, as 'paper/subject' is a component of a program. The courses should define learning activities, objectives and learning outcomes. Types of courses / activities constitute the programs of study comprise lectures outreach activities / Practical / Case Study / Group Discussion / Quiz / Project work/ Viva / Seminars / Assignment / Internship / Presentations / Research Project etc. or a combination of some of these.

4. **Major & Minor Discipline Course:** Major discipline is grouping of courses of main focus and the degree will be awarded in that discipline. Students should secure the prescribed number of credits (about 50% of total credits) through core courses in the major discipline.
   Minor discipline helps a student to gain a broader understanding beyond the major discipline.

5. **Multidisciplinary Courses:** These courses are intended to broaden the intellectual experience and form part of other disciplines.

6. **Ability Enhancement Courses:** The courses aim at enabling the students to acquire and demonstrate the core linguistic skills, including critical reading and expository and academic writing skills, that help students articulate their arguments and present their thinking clearly

and coherently and recognize the importance of language as a mediator of knowledge and identity.

7. **Skills Enhancement Courses:** These courses are aimed at imparting practical skills, hands-on training, soft skills, etc., to enhance the employability of students.

8. **Value-Added Courses / Indian Knowledge System:** The course aims at enabling the students to acquire and demonstrate the knowledge and understanding of contemporary India with its historical perspective, the basic framework of the goals and policies of national development, and the constitutional obligations with special emphasis on constitutional values and fundamental rights, ethics and duties.

9. **Summer Internship/ Apprenticeship:** Students will have to undergo Internships / Apprenticeships in a firm, industry, or organization or Training in labs with faculty and researchers in their own or other HEIs/research institutions during the summer term. Students will be provided with opportunities for internships with local industry, business organizations, health and allied areas, local governments (such as panchayats, municipalities). Parliament or elected representatives, media organizations, artists, crafts persons, and a wide variety of organizations so that students may actively engage with the practical side of their learning and, as a by-product, further improve their employability.

10. **Vocational Courses:** Vocational Education and Training will form an integral part of the undergraduate programme to impart skills along with theory and practical.

11. **Research Dissertation:** Students choosing a 4-Year Bachelor's degree (Honours with Research) are required to take up research dissertation under the guidance of a faculty member. The students are expected to complete the Research Dissertation in the eighth semester.

O. B.C.A. – 4:
Multiple Exit System:

1	**UG Certificate**	UG Certificate will be awarded when a student exits after completion of semester 1 and semester 2 with 44 credits along with successfully completion of *work based* vocational course of 4 credits Online/Offline or internship/Apprenticeship
2	**UG Diploma**	UG Diploma will be awarded when a student exits after completion of semester 1 to semester 4 with 88 credits along with successfully completion of *work based* vocational course of 4 credits Online/Offline or internship/Apprenticeship

3	**UG Degree**	Bachelor's degree will be awarded when a student exits after completion of semester 1 to semester 6 with 132 credits.
4	**UG Degree Honours / Honours with Research**	Bachelor's degree (Honours / Honours with Research) will be awarded when a student exits after completion of semester 1 to semester 8 with 176 credits.

O. B.C.A. – 5: Students are permitted to take a break or exit with a UG certificate / UG Diploma / UG Degree are permitted to re-enter within three years and complete the degree programme. But total duration for completing the programme shall not exceed 7 years.

O. B.C.A. – 6: No candidate will be admitted to any semester examination for B.C.A. unless it is certified by the principal that he/she has attended the course of study to the satisfaction of the principal of the college.

O. B.C.A. – 7: Candidate desirous of appearing at any semester examination of the B.C.A. programme must forward their application in the prescribed from to the University through the principal of the college on or before the date prescribed for the purpose under the relevant ordinances.

O. B.C.A. – 8: No candidate will be permitted to reappear at any semester examination, which he has already passed. The marks of successfully completed course will be carrying forwarded for the award of class.

O. B.C.A. – 9: Medium of instruction is English.

O.B.C.A. - 10: Any candidate can go up to take admission in successive semester irrespective of failure in any number of courses.

Regulations:

– 1. Standard Of Passing

The standard of passing the B.C.A. degree examination will be as under:

(1) To pass any semester examination of the B.C.A. degree, a candidate must obtain at least 40% marks in the examination separately in each course.

(2) Class will be awarded based on Earned Grade Point, SGPA and CGPA as per rules of university.

No.	Theory / Practical (SEE)	CCE	Result	Require to appear in Exam
1	PASS	PASS	PASS	N.A.
2	PASS	FAIL	FAIL	CCE ONLY
3	FAIL	PASS	FAIL	SEE ONLY
4	FAIL	FAIL	FAIL	BOTH (SEE & CCE)

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

– 2. Marks and credit hours of each course

(1) 4 Credit Theory Course:

- Total Marks of each theory course are 100 (SEE of 50 Marks + CCE of 50 Marks)
- Marks of Each Unit in the course are equal (i.e., 10 Marks). Total Marks of each course are 10x5=50 for SEE.
- Credit hours (lectures) for each unit in the course are equal (i.e., 12 hours). Total credit hours (lectures) of each course are 12x5 = 60.

(2) 4 Credit Practical / project-viva Course:

- Total Marks of each practical / project-viva course is 100 (SEE of 50 Marks + CCE of 50 Marks)
- Total Credit hours (practical) for this course is 120 hours.

(3) 2 Credit Course (AEC, IKS and SEC):

- Total marks for this course are 50 Marks (SEE of 25 Marks + CCE of 25 Marks)
- Total Credit hours for this course is 30 hours.

– 3. 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		Unit-1	03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following			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)			
	(b)	Attempt any one of the following		Unit-4	02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 5	(a)	Attempt the following			03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following		Unit-5	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)			

BCA SEM 1						
Sr. No.	**Type of Course**	**Course Title**	**Credit**	**CCE**	**SEE**	**Total**
1	MAJOR	CS-01: Problem Solving Methodologies and Programming In C	4	50	50	100
2	MAJOR	CS-02: Networking & Internet Environment	4	50	50	100
3	MINOR	CS-03: Computer Fundamentals and Emerging Technologies	4	50	50	100
4	MDC	CS-04: Practical Based on Problem Solving Methodologies and Programming In C and Networking & Internet Environment	4	50	50	100
5	AEC	CS-05: Critical Thinking and Problem Solving	2	25	25	50
6	SEC	CS-06: Mathematical and Statistical Foundation of Computer Science	2	25	25	50
7	IKS	CS-07: Mathematics in Ancient India: Exploring the Rich Heritage of Vedic Mathematics	2	25	25	50
			22	**300**	**250**	**550**

BCA SEM 2						
Sr. No.	**Type of Course**	**Course Title**	**Credit**	**CCE**	**SEE**	**Total**
1	MAJOR	CS-08: Data Structure Using C Language	4	50	50	100
2	MAJOR	CS-09: Web Programming	4	50	50	100
3	MINOR	CS-10: SAD, Software Quality Assurance & Testing	4	50	50	100
4	MDC	CS-11: Practical Based on Data Structure Using C Language and Web Programming	4	50	50	100
5	AEC	CS-12: Modern Indian Language	2	25	25	50
6	SEC	CS-13: Computer Organization & Architecture	2	25	25	50
7	VAC	CS-14: Environmental Science: Understanding the Earth's Ecosystems and Sustainability	2	25	25	50
			22	**300**	**250**	**550**

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

B.C.A. (Semester – 1)

Sr. No.	Type of Course	Course Title	Credit
1	**MAJOR**	**CS-01:** Problem Solving Methodologies and Programming In C	4
2	**MAJOR**	**CS-02:** Networking & Internet Environment	4
3	**MINOR**	**CS-03:** Computer Fundamentals and Emerging Technologies	4
4	**MDC**	**CS-04:** Practical Based on Problem Solving Methodologies and Programming In C and Networking & Internet Environment	4
5	**AEC**	**CS-05:** Critical Thinking and Problem Solving	2
6	**SEC**	**CS-06:** Mathematical and Statistical Foundation of Computer Science	2
7	**IKS**	**CS-07:** Mathematics in Ancient India: Exploring the Rich Heritage of Vedic Mathematics	2
		Total Credits of Semester 1	**22**

CS-01: PROBLEM SOLVING METHODOLOGIS AND PROGRAMMING IN C

Objectives:

- To develop basic programming skill and logic, concept of memory management and file handling.
- To be able to understand preprogramming techniques
- To become familiar with programming concepts
- To become familiar with different problem-solving methodologies

Prerequisites:

- Basic Computer Skills and Command-line knowledge

Unit No.	Topic	Detail
1	**Introduction of C Language**	• Introduction of Computer Languages • Introduction of Programming Concept • Introduction of C Language (History & Overview) • Difference between traditional and modern c. • C character set • C tokens ▪ Keywords ▪ Constants ▪ Strings ▪ Identifiers and variables ▪ Operators (all 8 operators) • Hierarchy of operators • Type casting • Data types in c • PRE-PROCESSORS IN C
	Introduction of Logic Development Tools	• Introduction of Logic. • Necessary Instructions for Developing Logic • Basics of Flow Chart • Dry-run and its Use. • Other Logic development techniques
2	**Control Structures**	• Selective control structure ▪ If statements ▪ Switch statement • Conditional ternary operator • Iterative (looping) control statements ▪ For loop ▪ Do…while loop ▪ While loop • Nesting of loops

		• Jumping statements ▪ Break, Continue and Goto statements
3	**Functions (Inbuilt and User Defined)**	• Types of library functions ▪ String Function: strcpy, strncpy, strcat, strncat, strchr, strrchr, strcmp, strncmp, strspn, strcspn, strlen, strpbrk, strstr, strtok ▪ Mathematical Functions: acos, asin, atan, ceil, cos, div, exp, fabs, floor, fmod, log, modf, pow, sin, sqrt ▪ I/O Formatting Functions: printf, scanf, getc, getchar, gets, putc, putchar, puts, ungetc ▪ Miscellaneous Functions: delay, clrscr, clearer, errno, isalnum, isalpha, isdigit, islower, isspace, isupper, isxdigit, toupper, tolower ▪ Standard Library functions: abs , atof , atol , exit , free, labs , rand , strtoul , srand ▪ Memory Allocation Functions: malloc , realloc , calloc • Types of user defined functions • Function call by value • Function call by reference • Recursion • Storage classes • Passing and returning values
4	**Array**	• Types of arrays ▪ Single dimensional array ▪ Two dimensional array ▪ Multi-dimensional array ▪ String arrays • Use of Arrays in Programming • Arrays and Matrices
	Pointers	• Introduction of Pointers • Use of pointers in Dynamic Programming • Pointer to Variables • Pointer to Array • Pointer within Array • Array of Pointer • Pointer To Structure • Pointers within structure • Pointer to Pointer • Dangling Pointer Problem
5	**User Defined Data Type – Structure, Union & enum**	• What is structure • Initializations and declarations • Memory allocation functions • Pointers with structures • Array with structures

		• User defined function with structures • Nested structures • Introduction to union • Difference between Structure & Union • Enumerated Type

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

Total Lectures 60 + 15 = 75

Reference Books:

1. Programming in C, by Pradip Dey & Manas Ghosh, Publisher – Oxford
2. C: The Complete Reference, by Herbert Schildt, Publisher – Tata McGraw Hill.
3. Programming in ANSI C Author : E. Balaguruswami.
4. Schaum's Outline of Programming with C, By: Byron Gottfried, Publisher Shaum Series
5. Programming with ANSI and Turbo C, by Ashok N Kamthane, Publisher – Pearson Education
6. Let Us C Author : Yashwant Kanetkar.
7. Working with C Author: Yashwant Kanitkar.

Course Outcome:

✓ Able to illustrate and explain basic concepts of programming
✓ Able to understand the concept of control statements.
✓ Able to translate the real-life situations in programming form and solve them using some fundamentals of Programming.
✓ Able to translate the real-life situations in programming form and solve them by storing data into files and analysed user defined data types and test and detect that it is optimized applications.

CS-02: NETWORKING & INTERNET ENVIRONMENT

Objectives:

- To give brief idea about Computer Network and Internet Environment
- To be able to design and develop static and/or interactive website using HTML5, CSS and Javascript.
- To become familiar with different web elements.
- To get intermediate knowledge of CSS3, Javascript and Bootstrap Framework

Prerequisites:

- Basic Knowledge of Computer Network and Web Surfing

Unit No.	Topic	Detail
1	**Introduction to Computer Network and it's Applications**	• Computer Network • Type of Computer Network • Different Terminologies used in Computer Network Internet, ISP (Internet Service Provider), Intranet, VSAT (very small aperture terminal), URL, Portal, Domain Name Server, World Wide Web (WWW), Search Engine, Remote Login, Telnet, Email, E-Commerce, E-Business, E-Governance, Mobile Commerce • Website Basics (WebPages; Hyper Text Transfer Protocol, File Transfer Protocol, Domain Names; URL; Protocol Address; Website[Static, Dynamic, Responsive etc], Web browser, Web Servers; Web Hosting
2	**Basic of HTML & Advance HTML 5**	• Fundamental of HTML • Basic Tag and Attribute • The Formatting Tags • The List Tags • Link Tag • inserting special characters, • adding images and Sound, • lists types of lists • Table in HTML • Frame in HTML • Forms • HTML 5 & Syntax o HTML5 Document Structure (section, article, aside, header, footer, nav,

		dialog, figure) o Attributes of HTML 5 o Web Form (datetime, date, month, week, time, number, range, email, url) o Audio / Video - Canvas
3	**Cascading Style Sheet & CSS 3**	• Introduction to CSS • Types of Style Sheets • Class & ID Selector • CSS Pseudo • CSS Font Properties • CSS Text Properties • CSS Background Properties • CSS List Properties • CSS Margin Properties • CSS Comments • CSS 3 o Border Property o Background & Gradient Property o Drop Shadow Property - 2D & 3D Transform Property o Transition Property o Box Sizing Property o Position Property • Media Query • CSS Flexbox Properties (display, flex-direction, flex-wrap, flex-flow, justify-content, align-items, align-content, gap row-gap, column-gap) • CSS Advance Properties (Overflow, text-overflow, Cursor, Visibility, filter, backdrop-filter, object-fit) • How to use Google Fonts & Custom Fonts (@font-face) • BEM Naming Convention
4	**Java Script**	• Introduction to JavaScript • Variables • JavaScript Operators • Conditional Statements • JavaScript Loops, Break and Continue Statements

		• Dialog Boxes
		• JavaScript Arrays • JavaScript User Define Function • Built in Function: string, Maths, Array, Date • Events (onclick, ondblclick, onmouseover, onmouseout, onkeypress, onkeyup, onfocus, onblur, onload, onchange, onsubmit, onreset) • DOM & History Object • Form Validation & E-mail Validation
5	**Bootstrap Framework**	• Introduction to Bootstrap • Bootstrap Layout (Container, Row, Columns, Responsive classes, Offset Column, Reordering Columns) • Bootstrap Content (Typography, Tables, Images, Forms) • Bootstrap Components (Navbar, Navs and tabs, Dropdowns, Buttons, Button Groups, Breadcrumb, Pagination, Labels, Alerts, Progress Bars, Accordion, Card, Modal) Bootstrap Utilities (Colors, Background, Borders, Display, Overflow, Position, Spacing, Text, Vertical align)

Seminar – 5 Lectures
Expert Talk – 5 Lectures
Test – 5 Lectures
Total Lectures: 60 + 15 = 75

Reference Books:

1. HTML in 10 steps or less - Laurie Ann Ulrich, Robert G. Fuller
2. Internet: The Complete Reference –Young.
3. World Wide Web Design with Html -C Xavier.
4. Internet for Every One –Leon.
5. Practical Html 4.O -Lee Philips.
6. MCSE Networking Essential Training Guides.
7. Benjamin Jakobus, Jason Marah, "Mastering BootStrap 4" 2nd Edition
8. Matt Lambert "Learning BootStrap 4", Packt Publishing

Course Outcome

✓ Able to understand Computer Network and Internet Environment
✓ Able to understand design and develop static and/or interactive website using HTML5, CSS and Javascript.
✓ Able to explore different web elements.
✓ Able to understand knowledge of CSS3, Javascript and Bootstrap Framework

CS-03: COMPUTER FUNDAMENTALS AND EMERGING TECHNOLOGY

Objectives:

- Bridge the fundamental concepts of computers with the present level of knowledge of the students.
- Familiarize peripheral devices, internal and external parts of computer system.
- Understand Number System like binary, hexadecimal and octal number systems and their arithmetic.

Prerequisites:

- Basic Computer Literacy

Unit No.	Topics	Details

1	**Introduction to Computers**	• Basics of Computers
		  o What is Computer?
		  o Characteristics of Computer
		  o Data Processing Cycle (Data → Process →information)
		• Classification of Computer by Data Processed
		  o Analog, Digital and Hybrid Computers
		• Classification of Computer by Processing Capabilities
		  o Micro, Mini, Mainframe and Super Computers
		• History and Generations of Computers
		  o First to Fifth Generation Computers
		• Simple Model of Computer
		  o Input Devices
		  o CPU (Central Processing Unit)
		  o Arithmetic & Logic Unit
		  o Control Unit
		  o Internal Memory
		• Output Devices
		• Secondary Storage Devices

	Internal/External parts used with Computer Cabinet	• Introduction to Mother board
		• Types of Processors.
		  o Dual Core, Core 2 Duo, i2, i3, etc ....
		• Memory structure and Types of Memory
		  o RAM (SRAM, DRAM, SO, DDR, etc.)
		  o ROM (ROM, PROM, EPROM, EEPROM, etc.)
		• Slots: ISA Slots / PCI Slots / Memory Slots
		• Sockets
		• Cables: Serial Cable / Parallel Cable / USB Cable

		• Ports: USB / Serial / Parellel / PS2 / HDMI • Power Devices: UPS • Graphic Cards, Network card, Sound Card

2	**Input Devices**	• Introduction
		• Types of Input Devices
		  ○ Keyboard / Mouse / Trackball / Glide - Pad / Game Devices Joystick, etc.) / Light Pen / Touch Screen / Digitizers and Graphic Tablet / Mic (Sound Input) / Camera (Photo and Video Input) / POS (Point of Sale) Terminal (Scanners, etc)
		  ○ MIDI(Musical Instrument Digital Interface) Keyboard,
		  ○ Wireless Devices (Keyboard, Mouse, etc)
		• Types of Scanners
		  ○ OCR, OMR, MICR, OBR

	Data Storage	• Introduction
		• Types of Magnetic Storage Devices
		  ○ Floppy Disk / Hard Disk (SATA, SSD) / Magnetic Tape / Magnetic Disks
		• Storage Mechanism of Magnetic Storage Devices
		  ○ Tracks / Sectors / Clusters / Cylinders
		• Reading / Writing Data to and from Storage Devices
		• Seek Time / Rotational Delay - Latency / Access
		• Time /Response Time
		• Other Storage Devices
		  ○ USB - Pen Drive / CD / DVD / Blu-Rav Disk etc.
		  ○ Flash Memory, Cloud Storage(Like Google Drive, OneDrive etc.)

3	**Output Devices**	• Types of Output Devices
		• CRT Display Units
		• Monitor
		• Non CRT display Units
		• LCD / LED / Plasma Displays
		• Types of Printers Impact and Non Impact Printers
		• Plotters
		• Other Devices
		  ○ Fascimile(FAX)
		  ○ OLED (Organic LED)
		  ○ Headphone
		  ○ SGD (Speech Generating Device)

		○ COM (Computer Output Microfilm) ○ Google Glass
4	**Numbering System and Codes**	• Introduction to Binary Codes / ○ Nibble / Bit / Byte / Carry Bit / Parity Bit / Sign Bit ○ KB / MB / GB / TB / HB (etc .... • Types of Numbering System ○ Binary / Octal/Decimal / Hex-Decimal • Conversion ○ Binary to Octal, Decimal and Hexa-Decimal ○ Decimal to Binary, Octal and Hexa-Decimal ○ Octal to Binary, Decimal and Hexa-Decimal ○ Hexa-Decimal to Binary, Octal and Decimal • Binary Arithmetic ○ Addition ○ Subtraction (1's Compliment and 2's Compliment) ○ Division ○ Multiplication ○ Types of Codes: ASCII/BCD / EBCDIC / Unicode • Parity Check: ○ Event Parity System / Odd Parity System
	Languages, Operating Systems and Software Packages	• Introduction • Translator (Assembler / Compiler / Interpreter) • Types of Languages ○ Machine Level Language ○ Assembly Level Language ○ High Level Language (3GL, 4GL, 5GL, etc.) • Types of Operating Systems ○ Batch Operating System ○ Multi Processing Operating System ○ Time Sharing Operating System ○ Online and Real Time Operating System • Uses and applications of Software Packages ○ Word Processing Packages ○ Spread Sheet Packages ○ Graphical Packages ○ Database Packages I ○ Presentation Packages ○ Animation / Video / Sound Packages

5	**Emerging Technologies and Virus**	• Different Communication methods
		  ○ GIS / GPS / CDMA / GSM
		• Communication Devices
		  ○ Cell Phones / Modem / Infrared / Bluetooth / WiFi/LiFi/SLM(Spatial Light Modulator)
		• Virus
		  ○ Introduction to Virus and related terms
		  ○ Origin and History
		  ○ Types of Virus
		  ○ Problems and Protection from Virus
		• Cloud Computing
		  ○ What is Cloud Computing?
		  ○ Characteristic & Service Models(Iaas, Paas, Saas)
		  ○ Architecture
		  ○ Security & Privacy

	Important Terms and Acronyms	• ATM • Backup / Restore • Hard Copy / Soft Copy • Bus / Data Bus • Buffer and types / Spooling • Cursor / Pointer / Icon • E-Mail I Attachment • CLil GUI • Compiler and its types • Drive I Directory (Folder) / File / Path • Menu / Popup Menu / Toolbar • Shutdown / Reboot / Restart • Syntax / Wild Card Characters • Optical Fiber (Fiber Optic) . • Net meeting • Printing Speed (CPS, CPM, LPM, DPI, PPM) • Peripherals

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

Total Lectures 60 + 15 = 75

Reference Books:

1. Computer Fundamentals – By P.K.Sinha.
2. Fundamental of IT for BCA – By S.Jaiswal.
3. Engineering Physics – By V.K.Gaur.
4. Teach Yourself Assembler – By Goodwin.

Course Outcome:

- ✓ Able to explore the fundamental concepts of computers
- ✓ Able to Understand peripheral devices, internal and external parts of computer system.
- ✓ Able to Understand Number System like binary, hexadecimal and octal number systems and their arithmetic.
- ✓ Able to recognize the emerging technologies
- ✓ Able to differentiate the types of virus

Additional Topics (Not to be asked in examination):

Student should be aware of followings

- To Format Hard Disk
- Installation of OS, multi-OS and other packages
- Use of DOS commands
- Operating of Accounting Software

CS-04 Practical based on CS-01 and CS-02	
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-04: Practical Based on Problem Solving Methodologies and Programming In C and Networking & Internet Environment	Total Marks - 100	
Topics	CCE	SEE
Problem Solving Methodologies and Programming in C	25	25
Networking and Internet Environment	25	25

Additional Topics to be taught during the semester – 1 (Not to be asked in examination):

Case studies of DBMS

CS-05: CRITICAL THINKING AND PROBLEM SOLVING

Objective:

- Identify and define problems clearly and accurately
- To use logic, reasoning and analytical tools to evaluate information
- To recognize the value of ongoing learning and reflection in problem-solving, and continuously work to improve skills and approaches.
- To generate creative and innovative solutions to complex problems, and evaluate potential outcomes and consequences.

Prerequisites:

- A willingness to engage in self-evaluation.

Unit No.	Topic	Details
1	**Personality Development**	• Self-awareness Conducting self-assessment exercises, personality tests. • Emotional Intelligence Practicing emotional regulation and social skills • Motivation Setting personal and academic goals and developing strategies to achieve them.
2	**Introduction to Critical Thinking and Problem Solving**	• Definition of critical thinking and problem solving • Importance of critical thinking and problem solving in personal and professional life • Approaches to critical thinking and problem solving • Techniques of problem solving
3	**Time Management and Goal Setting**	• Importance of time management • Techniques for managing time effectively • Goal setting and its importance • SMART goal setting • Prioritizing tasks

Course Outcome:

- ➢ Develop a deep understanding of critical thinking concepts.
- ➢ Develop the ability to identify and analyze problems critically, using logic and reasoning to evaluate different solutions and arrive at an effective decision.
- ➢ Enhance the ability to collaborate and communicate effectively with others, and work together to solve complex problems.

- Develop a creative mindset and an ability to think outside the box, and generate innovative solutions to complex problems.
- Develop the ability to learn from failure, and use these experiences to grow and improve problem-solving skills.

Reference Books:

- Thinking, Fast and Slow" by Daniel Kahneman
- "Critical Thinking: An Introduction to Analytical Reading and Reasoning" by Larry Wright
- "The Art of Thinking Clearly" by Rolf Dobelli
- "Critical Thinking: A User's Manual" by Debra Jackson and Paul Newberry

CS-06: MATHEMATICAL AND STATISTICAL FOUNDATION OF COMPUTER SCIENCE

Objectives:

- To create awareness of about basic Mathematics and Statistics
- To develop Reasoning ability, Logical ability and Arithmetic ability
- To develop a positive attitude towards learning Mathematics & statistics
- To perform mathematical & statistical operations and manipulations with confidence, speed and accuracy.

Prerequisites:

- Basic knowledge of Mathematics and Statistics

Unit No.	Topic	Details
1	**Determinants**	• Introduction • 2 × 2 , 3×3 order determinant • Cramer's method for solving linear equation (Two and Three Variables) • Properties of Determinants • Examples
	Matrices	• Introduction • Different types of matrix(square matrix, column matrix, row matrix, Diagonal matrix, Unit matrix, null matrix) • Transpose of matrix • Addition, subtraction & multiplication of two matrices • Adjoint of a square matrix • Inverse of matrix
2	**Measures of Central Tendency & Dispersion**	• Mean (ungroup data, group data) • Median (ungroup data, group data) • Mode (ungroup data, group data) • Range • Quartiles • Standard Deviation • Typical examples
3	**Arithmetic & Geometric progression**	• Sequence • Series • Arithmetic progression (Definition & Nth term, sum of n terms) • Geometric progression (Definition & Nth term, sum of n terms) • Harmonic Progression • Relation Between AM GM HM (Two Numbers) • Typical examples

Student Seminar – 5 Lectures
Expert Talk – 5 Lectures
Student Test – 5 Lectures
Total Lectures 30 + 15 = 45

Course Outcome:

- Able to Understand basics of Mathematics and Statistics
- Able to Develop reasoning ability, logical ability and arithmetic ability
- Able to Develop a positive attitude towards learning Mathematics & statistics
- Able to Perform mathematical & statistical operations and manipulations with accuracy.

Reference Books:

1. Business Mathematics By Sancheti & Kapoor Sultan & Chand
2. Statistical Method By Gupta Sultan & Chand
3. Discrete Mathematical Structures with Applications to Computer Science By J.P. Tremblay & R. Manohar TMH
4. Business Mathematics : V.K. Kapoor
5. Business Mathematics : Dr Kachot
6. Fundamentals of Statistics : S. C. Gupta

CS-07: Mathematics in Ancient India: Exploring the Rich Heritage of Vedic Mathematics

Objectives:

- Helps students understand the contributions made by ancient civilizations to the field of mathematics.
- Ancient mathematics helps to establish connections between past and present mathematical ideas.
- Exploring Mathematical concepts.

Prerequisites:

- Eagerness to know rich heritage of Indian Mathematics.

Unit No.	Topic	Details
1	**Biographies of Ancient Indian Mathematicians**	• A brief introduction to the lives and information on the works of the following mathematicians: Aryabhata, Varahmihira, Brahmagupta, Bhaskara I & II
2	**Biographies of Remarkable Indian Mathematicians**	• A brief introduction to the lives and information on the works of the following mathematicians: Shrinivasa Ramanujan, C. R. Rao, P. C. Mahalanobis, D. R. Kaprekar, Satyendranath Bose, Shakuntala Devi
3	**Vedic Mathematics and Mathematics**	• Overview of Vedic Mathematics and its historical background. • Introduction to the 16 Sutras and 13 Upa-Sutras used in Vedic Mathematics. • Use of Vedic Mathematics • Importance of Vedic Mathematics

Course Outcome:

- ➢ Student will know the Mathematical advancements of Ancient India.
- ➢ Student will gain a deeper understanding of the historical development of mathematics in ancient civilizations.
- ➢ Enhance their problem-solving skills and discovering the connections between ancient mathematical ideas and modern mathematical concepts.

Reference Books:

- The History of Ancient Indian Mathematics. The World Press Private Ltd. Calcutta. Digitized Book (2009) - Srinivasiengar, C. N. (1988).
- "Vedic Mathematics" by Swami Bharati Krishna Tirtha

BCA (Semester – 2)

Sr. No.	Type of Course	Course Title	Credit
1	MAJOR	CS-08: Data Structure Using C Language	4
2	MAJOR	CS-09: Web Programming	4
3	MINOR	CS-10: SAD, Software Quality Assurance & Testing	4
4	MDC	CS-11: Practical Based on Data Structure Using C Language & Web Programming	4
5	AEC	CS-12: Modern Indian Language	2
6	SEC	CS-13: Computer Organization & Architecture	2
7	VAC	CS-14: Environmental Science: Understanding the Earth's Ecosystems and Sustainability	2
		Total Credit of Semester - 2	**22**

CS-08: DATA STRUCTURE USING C LANGUAGE

Objectives:

- To provide the knowledge of basic data structures and their implementations.
- To understand importance of data structures in context of writing efficient programs.
- To develop skills to apply appropriate data structures in problem solving

Prerequisites:

- Computer Programming Knowledge
- Fundamental knowledge of C Programming

Sr. No.	Topic	Detail
1	**Algorithm Analysis**	• The analysis of algorithm. • Time and space complexities. • Asymptotic notation. • Classes of algorithm. • Big-Oh Notation • Big-Omega Notation
	File Handling	• Concept of data files • File handling • Use of file handling functions fopen, fclose, fprintf, fscanf, getw, putw, fseek, ftell, rewind ,freopen, remove, rename, feof, ferror • I/O operations • Command line arguments
2	**Sorting and Searching**	• Bubble sorting • Insertion sorting • Quick sorting • Bucket sorting • Merge sorting • Selection sorting • Shell sorting • Basic searching technique: Index searching, Sequential searching, Binary searching
3	**Introduction To data Structure**	Primitive and simple structures Linear and nonlinear structures file organization.
	Elementary Data Structure	• Stack o Definition o Operations on stack

		○ Implementation of stacks using arrays ▪ Function to insert an element into the stack ▪ Function to delete an element from the stack ▪ Function to display the items ○ Recursion and stacks ○ Evaluation of expressions using stacks ▪ Postfix expressions ▪ Prefix expression • Queue ○ Introduction ○ Array implementation of queues ○ Function to insert an element into the queue ○ Function to delete an element from the queue • Circular queue ○ Function to insert an element into the queue ○ Function for deletion from circular queue ○ Circular queue with array implementation • Deques • Priority queues
4	**Linked List & Implementation**	• Applications of the linked lists • Types of Linked Lists ○ Singly Linked List ○ Doubly linked list ○ Header Linked List ○ Circular Linked List • Implementation using Singly Linked List, Doubly Linked List and Circular Singly Linked List ○ Insertion of a node at the beginning ○ Insertion of a node at the end ○ Insertion of a node after a specified node ○ Traversing the entire linked list ○ Deletion of a node from linked list ○ Updating of a specific node • Implementation of merging of two Singly Linked List • Implementation of reversing of Singly Linked List
5	**Tree**	• Objectives • Properties of a tree • Binary trees ○ Properties of binary trees ○ Implementation ○ Traversals of a binary tree

		▪ In order traversal ▪ Post order traversal ▪ Preorder traversal • Binary search trees (bst) o Insertion in bst o Deletion of a node o Search for a key in bst • Height balanced tree • B-tree Algorithm o Insertion, Deletion
	Graph	• Adjacency matrix and adjacency lists • Graph traversal o Depth First Search (DFS) o Implementation o Breadth First Search (BFS) o Implementation • Shortest path problem • Minimal spanning tree

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures
Total Lectures 60 + 15 = 75

Reference Books:

1. Data Structure through C/C++ Author : Tennaunbuam.
2. Let us C Author : Kanitkar.
3. Pointer in C Author : Kanitkar.
4. Data and File Structure Author : Trembley & Sorrenson.

Course Outcome:

- Able to Understand basic data structures and their implementations.
- Able to Understand importance of data structures in context of writing efficient programs.
- Able to Develop skills to apply appropriate data structures in problem solving
- Able to Explore tree and graph data structure

Additional Topics to be taught during the semester – 2 (Not to be asked in examination):

- Case studies of data structure

CS-09: WEB PROGRAMMING

Objectives:

- To create dynamic website / web based applications using PHP - MySQL Database.
- Able to develop website with the use of jQuery, AJAX and JSON.
- To become familiar with OOPs concept.

Prerequisites: Basic knowledge of Programming

Unit No.	Topic	Detail
1	**PHP Basic**	• Introduction to PHP • PHP configuration in IIS & Apache Web server • Understanding of PHP.INI file • Understanding of PHP .htaccess file • PHP Variable • Static & global variable • GET & POST method • PHP Operator • Conditional Structure & Looping Structure • Array • User Defined Functions: ▪ argument function ▪ default argument ▪ variable function ▪ return function • Variable Length Argument Function ▪ func_num_args ▪ func_get_arg, func_get_args • Built in Functions - Variable Functions - String Function - Math Function - Date Function - Array Function - Miscellaneous Function - File handling Function
2	**Handling Form, Session Tracking & PHP Components**	• Handling form with GET & POST • Cookies • Session • Server variable • PHP Components

		- PHP GD Library - PHP Regular expression - Uploading file - Sending mail
	AJAX & JSON	• What is AJAX? • PHP with AJAX • MySql with AJAX • What is JQuery AJAX • JQuery AJAX with PHP • Introduction to JSON o Installation & Configuration o Resource Types o JsonSerializable o JSON Functions: json_decode, json_encode
3	**Introduction of SQL**	• Working with MySQL using PhpMyAdmin • SQL DML Statement (Insert, Update, Select, Delete) Command • PHP-MySQLi Connectivity • PHP-MySQLi Functions • mysqli_connect, mysqli_close,mysqli_error, msyqli_errno, mysqli_select_db, mysqli_query, mysqli_fetch_array, mysqli_num_Rows, mysqli_affe cted_Rows, mysqli_fetch_assoc, mysqli_fetch_field , mysqli_fetch_object,mysqli_fetch_row, mysqli_insert_id, mysqli_num_fields, mysqli_data_seek
4	**jQuery**	What is jQuery? jQuery Syntax jQuery Selector - Element Selector - Class Selector - id Selector jQuery Events: Click, dbclick, keypress, keydown, keyup, submit, change, focus, blur, load, resize, scroll, unlode jQuery Effects: hide show, fade, slide jQuery Methods: css, height, width, innerWidth, innerHeight, outerWidth, outerHeight, html, text, append, prepend, after, before, addClass, removeClass, toggleClass, remove, empty
5	**OOP**	• Concept of OOP o Class o Object o Property o Visibility

		○ Constructor, Destructor ○ Inheritance ○ Scope Resolution Operator (::) ○ Autoloading Classes ○ Class Constants • Mysql Database handling with oop (insert, update, select, delete

Seminar - 5 Lectures
Expert Talk - 5 Lectures
Test - 5 Lectures
Total Lectures: 60+15=75

Reference Books:

1. Modern PHP: New Features and Good Practices by Josh Lockhart (ORELLY)
2. PHP Cookbook: Solutions & Examples for PHP Programmers by David Sklar and Adam Trachtenberg (ORELLY)
3. Programming PHP by Kevin Tatroe and Peter MacIntyre ORELLY)
4. PHP for the Web: Visual QuickStart Guide (4th Edition) by Larry Ullman (Peachpit Press)

Course Outcome:

- Able to Understand Creation of dynamic website / web-based applications using PHP - MySQL Database.
- Able to Understand development of website with the use of jQuery, AJAX and JSON.
- Able to Understand practical and real-life examples of OOP.

Additional Topics (Not to be asked in examination):

Student should be aware of followings

- Case Study
- Uses and Advantages of CMS
- Wordpress [Introduction & Installation]
- Joomla [Introduction & Installation]
- Magento [Introduction & Installation]

CS – 10: SAD, Software Quality Assurance and Testing

Objectives:

- To Understand and explore concept of System Analysis
- To Understand concept of System Development Life Cycle
- To Understand Quality Assurance
- To Understand concept of Software Testing
- To explore the concept of Project Tracking and Scheduling
- To Understand the concept of Quality Control and Testing
- To Understand the software models and Automated Testing
- To Understand the UML Diagram
- To Understand the concept of CAD Project Management

Prerequisites:

- Problem-Solving Skills
- Basic concepts of Database
- Basic knowledge of Software Development Fundamentals

Unit No.	Topics	Details
1	**System Analysis & Design, Software Engineering & Concept of Quality Assurance**	Definitions: System, Subsystem, Business System, Information System (Definitions only) Systems Analyst (Role: Information Analyst, Systems Designer & Programmer Analyst) SDLC • Fact – finding techniques (Interview, Questionnaire, Record review and observation) Tools for Documenting Procedures and Decisions Decision Trees and Decision Tables Data Flow analysis Tool DFD (context and zero level) and Data Dictionary Software Engineering (Brief introduction) Introduction to QA Quality Control (QC) Difference between QA and Q Quality Assurance activities

2	**Basics of Software Testing, Types of Software Testing, Verification and Validation**	• Introduction to software Testing
		• Software faults and failures
		  • Bug/Error/Defect/Faults/Failures
		• Testing Artifacts
		  • Test case
		  • Test Script
		  • Test Plan
		  • Test Harness
		  • Test Suite
		• Static Testing
		  • Informal Review
		  • Walthrough
		  • Technical Review
		  • Inspection
		• Dynamic Testing
		• Test levels
		  • Unit Testing
		  • Integration Testing
		  • System Testing
		  • Acceptance Testing
		Techniques of software Testing
		• Black Box Testing
		  • Equivalence Partitioning
		  • Boundary Data Analysis
		  • Decision Table Testing
		  • State Transition Testing
		• White Box Testing
		  • Statement testing and coverage
		  • Decision testing and coverage
		• Grey Box Testing
		• Nonfunctional Testing
		  • Performance Testing
		  • Stress Testing
		  • Load Testing
		  • Usability Testing
		  • Security Testing

3	**Software Development Life Cycle Models and Automated Testing**	• Waterfall Model
		• Iterative Model
		• V-Model
		• Spiral Model
		• Big Bang Model
		• Prototyping Model
		• Introduction to Automated Testing
		  • Concept of Freeware, Shareware, licensed tools
		• Theory and Practical Case-Study of Testing Tools
		  • Selenium
		  • Neoload
		  • Junit
		  • Nunit
		  • Acunetix
		  • ZAP

4	**Project Economics, Project scheduling and Tracking**	• Concepts of Project Management • Project Costing based on metrics • Empirical Project Estimation Techniques. • Decomposition Techniques. • Algorithmic methods. • Automated Estimation Tools • Concepts of project scheduling and tracking • Effort estimation techniques • Task network and scheduling methods • Timeline chart • Pert Chart • Monitoring and control progress • Graphical Reporting Tools

5	**CAD Project Management Tool UML**	• MS – VISIO for designing & Documentation • MS – Project for controlling and Project Management • UML designing and skill based tools Overview of ♦Class Diagram ♦Use Case Diagram ♦Activity Diagram

Student seminar - 5 Lectures.
Expert Talk - 5 Lectures
Students Test - 5 Lectures.
TOTAL LECTURES 60+15=75

Reference Books

1. Analysis & Design of Information System - James A. Senn.
2. Pankaj Jalote, "Software Engineering – A Precise Approach", Wiley India
3. UML Distilled by Martin Fowler, Pearson Edition, 3rd Edition
4. Fundamentals of Software Engineering – RajibMall (PHP)
5. Software Engineering – A Practitioner's Approach – Pressman
6. UML – A Beginner's Guide –Jasson Roff – TMH
7. Roger Pressman , "Software Engineering"
8. http://en.wikipedia.org/wiki/Software_testing
9. http://www.onestoptesting.com/
10. http://www.opensourcetesting.org/functional.php

Course Outcome

- Able to Understand and explore concept of System Analysis
- Able to Understand concept of System Development Life Cycle
- Able to Understand Quality Assurance
- Able to Understand concept of Software Testing
- Able to Explore the concept of Project Tracking and Scheduling
- Able to Understand the concept of Quality Control and Testing
- Able to Understand the software models and Automated Testing
- Able to Understand the UML Diagram
- Able to Understand the concept of CAD Project Management

CS-11 Practical based on CS-08 and CS-09	
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-11: Practical Based on Data Structure Using C Language & Web Programming	Total Marks - 100	
Topics	CCE	SEE
Data Structure using C language	25	25
Web Programming	25	25

CS-12: MODERN INDIAN LANGUAGE

Objective:

- To enable students to develop basic proficiency in reading, writing, speaking and listening in the target language.
- To introduce students to the script or writing system used in the language.

Prerequisites:

- Knowledge of communication and interpersonal skills
- Interest to develop the language proficiency

Unit No.	Topic	Details
1	**Language Practice**	• Role-playing exercises • Group Discussion and presentation • Language games and activities
2	**Language Proficiency**	• Reading comprehension and speed • Speaking and Listening skills • Technical and academic vocabulary

3	**Activities**	• Creating
		  ○ Short-Story
		  ○ Poem
		  ○ Dialog

Course Outcome:

➢ Students will develop basic communication skills in the target language, including reading, writing, speaking and listening.

➢ Students will develop an appreciation for the language and its cultural significance.

Reference Books:

➢ Language Practice By Michael Vince – Published By macmillan

➢ The Handbook of Advanced Proficiency in Second Language Acquisition – Editors: Alessandra G. Benati, Paul A. Malovrth – Published By Willey

CS-13: COMPUTER ORGANIZATION AND ARCHITECTURE

Objectives:

- Understand how logic circuits and boolean algebra forms as the basics of digital computer.
- Demonstrate the building up of Sequential and Combinational logic from basic gates

Prerequisites:

- General Knowledge of Computer

Unit No.	Topic	Detail
1	**Digital Logic Circuits**	• **Logic Gates** ▪ AND,OR,NOT,NAND,NOR,XOR, Exclusive NOR gates • **Boolean Algebra** ▪ Boolean algebra? ▪ Boolean variable and Boolean function (Analog and Digital Signals) ▪ Truth table ▪ Postulates ▪ Theorem related to postulates ▪ Simplified Boolean function using postulates and draw logical diagram of simplified function ▪ Simplified Boolean function using Karnaugh map method with DON'T CARE condition • **Sequential And Combinational Circuits** ▪ Clock pulses ▪ Combinational circuit, sequential circuit and adder • **Flip Flops** ▪ SR, Clocked SR, D, JK, JK – Master Slave, T • **Universal Gate**
2	**Central Processing Unit**	• Introduction Of CPU • Major component of CPU • General Register Organization ▪ control word ▪ Accumulator Register • **Stack Organization** ▪ Register stack ▪ Memory stack ▪ Polish notation and reverse polish notation

		• **Arithmetic And Logic Unit** ▪ Block diagram of ALU • **Interrupts**
3	**Input-Output Organization**	• Memory buses • Block diagram and function • Data Bus, Address Bus and Control lines • Input Output Buses • Concept of input output interface • Input Out Processor (IOP) • Direct Memory Access • DMA controller

Student seminar - 5 Lectures
Expert Talk - 5 Lectures
Students Test - 5 Lectures
Total Lectures 60 + 15 = 75

Reference Books:

1. Computer System Architecture – By Morris Mano (PHI).
2. Digital Logic And Computer Design – By Morris Mano.
3. Digital Computer Electronics – By Malvino And Leach.

Course Outcome:

- Able to Understand logic circuits and boolean algebra forms as the basics of digital computer.
- Able to Explore the building up of Sequential and Combinational logic from basic gates
- Able to explore digital components
- Able to Understand data representation

Hands On (Not to be asked in examination):

- Instruction Formats - Simulator Base Program

Additional Topics to be taught during the semester-2 (Not to be asked in examination):

Following tools should be used to train students.

- Simulator 8051
- Using Trainer kit

CS-14: Environmental Science: Understanding the Earth's Ecosystems and Sustainability

Objective:

- The primary objective is to introduce students to the fundamental concepts of Environmental Science, including ecosystems, biodiversity, natural resources, pollution, climate change, and sustainability.
- The course aims to raise awareness about pressing environmental challenges faced globally and locally, such as air and water pollution, deforestation, habitat destruction, and climate change.
- Students will become familiar with environmental laws, regulations, and policies at local, national, and international levels, which govern environmental protection and conservation efforts.

Prerequisites:

- A fundamental understanding of basic science subjects.

Unit No.	Topic	Details
1	**Introduction to Environment Science**	• Definition • Environmental Issues and Challenges • Principles and Scope • Concepts of Ecology and Ecosystem
2	**Environmental Pollution**	• Types of Pollution (air, water, soil, noise, etc.) • Sources and impact of pollution • Mitigation and control measures
3	**Climate Change and Global Warming**	• Greenhouse effect and its implications • Causes and consequences of climate change • Sustainable practices to combat global warming

Course Outcome:

➢ Students will demonstrate a solid understanding of environmental concepts.

➢ Students will develop an increased awareness of pressing environmental issues facing the planet today and recognize the interconnections between human activities and the environment.

Reference Books:

➢ "Environmental Science" by G. Tyler Miller and Scott Spoolman

➢ Environmental Impact assessment – L W Canter – McGraw Hill

BCA-1 **CS-01: Problem solving methodologies and programming in C** **Minimum following exercise should be performed by the students during the semester**
1. Write a C program to display "Hello, World!" on the screen.
2. Write a C program to display your name, age, and city using printf().
3. Write a C program to Print Your Name in Center of the screen
4. Write a C program to Print Your Name in Four corners of the screen
5. Write a C program to declare a variable and assign value and display that variable.
6. Write a C program to get integer value and display it
7. Write a C program to perform addition, subtraction, multiplication, and division of two integers.
8. Write a C program to find the square and cube of a number.
9. Write a C program to get 5 numbers from user, display sum and average.
10. Write a C program to input two numbers and swap them using a third variable.
11. Write a C program to input two numbers and swap them without using a third variable.
12. Write a C program to calculate simple interest. (SI = P * R * T / 100)
13. Write a program to calculate compound interest.
14. Write a C program to convert temperature from Celsius to Fahrenheit.
15. Write a C program to Input meters and convert it into centimeters
16. Write a C program to Input feet's and convert it into inches.
17. Write a C program to convert inches to centimeters
18. Write a C program to input distance in kilometers and convert it into meters, centimeters, and millimeters.
19. Write a C program to calculate the area of a circle. (Area = π * r * r)
20. Write a C program to calculate the volume of a cylinder. (Volume = π * r^2 * h)
21. Write a C program to input length and width of a rectangle and calculate area and perimeter.
22. Write a program to calculate area and perimeter of triangle.
23. Write a program to input seconds and convert into HH:MM:SS format.
24. Write a program to convert days into years, weeks
25. Write a C program to print the ASCII value of a character entered by the user.
26. Write a C program to perform type casting from float to int and vice versa.
27. Write a C program to use the sizeof operator to display memory size of basic data types.
28. Write a C program to calculate the gross salary of an employee (HRA = 20%, DA = 40% TAX=3% of basic salary).
29. Write a C program to calculate 5% discount amount and final amount for a product purchase by customer.
30. Write a C program to define a constant value using #define and calculate the area of a

circle.
31. Write a C program to input a product price and quantity, then calculate the total cost and 5% tax. Display payable amount.
32. Write a C program to input three item prices and calculate the total shopping bill including a fixed 5% GST.
33. Write a c program to input working days and per day salary. Calculate basic salary and 10% bonus on it. Display net salary
Condition programs
1. Write a program to check whether a number is positive, negative, or zero.
2. Write a C program to check whether a number is even or odd using if-else.
3. Write a C program to find the larger of two numbers using if-else.
4. Write a C program to find the largest among three numbers using else-if.
5. Write a C Program to find the middle value out of three values
6. Write a C program to check whether a year is a leap year or not.
7. Write a program to check number is 2 digit or not.
8. Write a program to display absolute value of given number
9. Write a program to convert negative number to positive and positive to negative.
10. Write a program to input age and check if person is child, teenager, adult, or senior**.**
11. Write a program to check if a number is divisible by 5 and 11.
12. Write a program to find the eligibility of a person to vote based on age.
13. Write a C program to implement a simple calculator using switch-case.
14. Use switch-case to display day of the week (1-7 input).
15. Use switch-case to display month name (1-12 input).
16. Switch-case to calculate area of different shapes (circle, rectangle, triangle) based on user's choice.
17. Write a program to calculate the total, percentage, result and grade of a student based on marks of 5 subjects (using if-else ladder).
18. Write a program to calculate the discount on a product: If the price of the product is more than ₹1000, apply a 10% discount; otherwise, apply a 5% discount. Display the final price after discount.
19. Write a program to calculate income tax based on income slabs:
a. Income up to ₹2,50,000: No tax
b. Income from ₹2,50,001 to ₹5,00,000: 5% tax on the amount above ₹2,50,000
c. Income from ₹5,00,001 to ₹10,00,000: 20% tax on the amount above ₹5,00,000
d. Income above ₹10,00,000: 30% tax on the amount above ₹10,00,000 Calculate and display the total tax.
20. Write a program to calculate the salary of an employee: Basic salary is given. Add 20% house rent allowance (HRA) and 10% dearness allowance (DA). Deduct 12% provident fund (PF). Calculate and display net salary.
21. Write a program to calculate electricity bill based on units consumed:
a. First 100 units: ₹1.50 per unit

b. Next 100 units: ₹2.50 per unit
c. Beyond 200 units: ₹3.50 per unit Calculate and display the total bill amount.
22. Write a program to calculate total price after discount and GST: Apply 8% discount if purchase amount exceeds ₹5000. Add 18% GST on the discounted amount. Display the final amount.
23. Writeamenudriveprogramwhichhasoptionsasfollow:
i. PrintAsciiofenteredcharacter
ii. Printtableofenterednumber
iii. PrintOdd oreven
24. Writeaprogramthatreadsastudent'sscore(anintegerbetween0and100)determine the corresponding letter grade based on the following scale:
a. A: 90-100
b. B:80-89
c. C:70-79
d. D: 60-69
e. F: 0-59
Theprogramshouldthenprintthelettergrade
25. Createaprogramthatreadstheageofapersonandclassify themintooneof the following age groups:
a. Child(0-12years)
b. Teenager(13-19years)
c. Adult(20-64 years)
d. Senior(65yearsandolder)
Print the age group category
26. Createaprogramthatreadstheweight(inkilograms)andheight(inmeters)ofa person and to categorize their Body Mass Index (BMI) as follows:
a. Underweight:BMI< 18.5
b. Normalweight:18.5≤ BMI<24.9
c. Overweight:25≤ BMI< 29.9
d. Obese: BMI≥30
Printthecorrespondingcategory.(ConsiderBMI=WEIGHT/HEIGHT)
27. Design a program that reads a person's income and credit score to determine their eligibility for a loan. Use if and else if statements with the following criteria:
a. Income≥50,000andCreditScore≥700:Eligibleforloan
b. Income≥30,000andCreditScore≥650:Conditionaleligibility
c. Otherwise:Noteligible Print the eligibility status.
Character programs
1. Write a program to find the ASCII value of a given character.
2. Write a program to check whether a character is a vowel or a consonant.
3. Check if two characters are equal or not.
4. Write a program to determine if a character is uppercase or lowercase.
5. Write a program to check whether a character is an alphabet, digit, or special

character.
6. Write a program to count number of uppercase, lowercase, digits, and special characters in a string.
7. **Write a program to convert an uppercase character to lowercase and lower case** to upper case
Loop programs
1. Write a C program to print numbers from 1 to N using for loop.
2. Write a program to print even numbers from 1 to N.
3. Write a program to print odd numbers from 1 to N.
4. Print numbers from N to 1 in reverse
5. Print the series: 1 4 9 16 25 ... N terms (squares of natural numbers)
6. Print the series: 1 8 27 64 125 ... N terms (cubes of natural numbers)
7. Print 100 81 64…n
8. Print 1 2 6 24 120……n
9. Print the series: **1 2 6 24 120 ... N terms** (factorials)
10. Print the series: **2 5 10 17 26 37 ... N terms** ($n^2 + 1$)
11. Print the series: **1 2 4 7 11 16 22 ... N terms** (difference increases by 1)
12. Print the series: **1, -2, 3, -4, 5, ... N terms** (alternating signs)
13. Print the series: **1 - 2 + 3 - 4 + 5 - ... ± N**
14. Print the series: **1 - 2 + 3 - 4 + 5 - ... ± N = ?**
15. Print the series: **1 + 1/2 + 1/3 + ... + 1/N** (harmonic series)
16. Print the series: **1 - 1/2 + 1/3 - 1/4 + ... ± 1/N**
17. Print the series: **2, 4, 8, 16, 32, ... (2^n)**
18. Print the series: **0, 3, 8, 15, 24, 35, ... ($n^2 - 1$)**
19. Print the series: **1×2, 2×3, 3×4, ..., N×(N+1)**
20. Print the series: 10 1 9 2 8 3 7 4 6 5
21. Print the series: **$1^2 + 2^2 + 3^2 + ... + N^2$**
22. Print the series: **$1^3 + 2^3 + 3^3 + ... + N^3$**
23. Print the series: **1 + 10 + 100 + 1000 + ... N terms**
24. Print A a B b C c D d ..... Z z
25. Write a program to calculate sum of even and odd numbers up to N.
26. Calculate and print sum of series of numbers which are divisible by 3
27. Write a C program to find the sum of first N natural numbers using loop.
28. Write a C program to find the factorial of a number using loop.
29. Write a C program to display the multiplication table of a number entered by the user.
30. Write a C program to check whether a number is prime or not.
31. Write a program to print the Fibonacci series up to N terms using a loop.

32. Write a C program to count the number of digits in an integer.
33. Write a C program to reverse a given integer number.
34. Write a C program to find the sum of digits of a given number.
35. Write a C program to check whether a number is an Armstrong number.
36. Write a C program to check whether a number is a palindrome
Perform Followings with the use of for loop, while loop, do ….while loop and goto statement.

##### ##### ##### ##### #####	1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5	1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5	5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1
11111 ##### 11111 ##### 11111	1111# 111## 11### 1#### #####	#$#$# #$#$# #$#$# #$#$# #$#$#	55555 44444 $$$$$ 22222 11111
1 12 123 1234 12345	1 22 333 4444 55555	5 54 543 5432 54321	5 44 333 2222 11111
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1 00 1111 00000 11111	11111 2222 333 44 5	12345 2345 345 45 5
10101 1010 101 10 1	5 54 543 5432 54321	12345 1234 123 12 1	55555 4444 333 22 1
* * * * * * * * * * * * * * *	* * * * * * * * * * * * * * *	1 121 12321 1234321 123454321	123454321 1234321 12321 121 1

A BB CCC DDDD EEEEE	A BC DEF GHI JKLM	ABCDE ABCD ABC AB A	&&&&& ###### &&&&& ###### &&&&&
* *** ***** ******* *********	********* ******* ***** *** *	1 232 34543 4567654 567898765	* * * * * * * *****
***** * * * * * * *****	1 2 2 3 3 3 2 2 1	* - * - * - * - * - * - * - * * - * - * * - * - *	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1
1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	1 0 1 1 0 1 0 1 0 1 0 1 0 1 0	2 4 6 8 10 12 14 16 18 20 22 24 26 28 30	1 2 3 4 5 1 2 3 4 1 2 3 1 2 1
1 00 111 0000 11111	55555 54444 54333 54322 54321	\ ‘ ‘ ~~~ ==== >>>>>	A 1 B 2 C3 D4E5 F 6G7H
1A2B3C4D5E 1A2B3C4D 1A2B3C 1A2B 1A	********** **** **** *** *** ** ** * *	1 121 12321 1234321 123454321	* * ** ** *** *** **** **** **********
##### #$$$# #$$$# #$$$# #####	AAAAA BBBBB 11111 CCCC DDDD	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
* *** ***** ******* ********* ******* ***** ***	1 121 12321 1234321 123454321 1234321 12321 121	* *	1 1 12 21 123 3 21 1234

*	1	* * * * * * * * * * * * * * * * * * * * *	43 21 12345 5 43 2 1

One Dimensional Array	
(1)	Write program to get 10 numbers and display it
(2)	Write program to display number with index
(3)	Write program to sum of array.
(4)	Write a program to display array in reverse order.
(5)	Write a program to find out maximum number.
(6)	Write a program to find minimum number from array.
(7)	Write a program to sort array.
(8)	Write a program to search number from array and display its position.
(9)	Write a program to display sum of odd and even numbers
(10)	Write a program to count odd and even number form the array
(11)	Input 5 numbers and count how many of them are positive, negative and zeroes
Two dimensional Array	
(1)	Write a program to print sum of 3 X 3 matrix
(2)	Write a program to print multiplication of 3 X 3 matrix
(3)	Write a program for 3 X 3 matrix transpose
(4)	Write a program to display sum row wise and column wise
(5)	Write a program to get five names and display it in upper case.
String Handling	
(1)	Write a program to display each character in new line with ASCII value (i.e. Entered string: “Computer” o/p: C – 67, o – 111, m – 109, p – 112, u – 117, t – 116, e – 101, r – 114)
(2)	Write a program to find character from the string
(3)	Write a program to count vowels from the string
(4)	Write a program to display string in reverse
(5)	Write a program to display length of string.
(6)	Write a program to Calculate length of a string without using strlen()
(7)	Write a program that accepts STRING from keyboard and count the character is a CAPITAL letter, a SMALL letter, a DIGIT or a SPECIAL character.
(8)	Write a program to print characters between two given characters (e.g., from 'A' to 'Z').
(9)	Print uppercase and lowercase alphabets in alternating order (A a B b ...).
(10)	Write a program that reads a string and print each word in new line.
(11)	Write a program to compare two strings
(12)	Write a program to check string is palindrome or not.
(13)	Write a program to reverse a string using character indexing.
(14)	Write a program to compare two strings without using strcmp().
(15)	Input a name and print each character with toggle case.
UDF / function	

(1)	Write a C program to copy one string to another using strcpy().
(2)	Write a program to concatenate two strings using strcat().
(3)	Write a C program to Create a program to compare two strings using strcmp().
(4)	Write a C program to Use strlen() to find the length of a string.
(5)	Write a C program to Search for a character in a string using strchr() and strrchr().
(6)	Write a C program to Use strstr() to find a substring inside a string.
(7)	Write a C program to Tokenize a sentence using strtok() function.
(8)	Write a C program to Compare first n characters of strings using strncmp().
(9)	Write a C program to Copy first n characters of a string using strncpy().
(10)	Write a C program to Use strspn() and strcspn() to analyze string segments.
(11)	Write a C program to Use strpbrk() to find the first occurrence of any character from a set.
(12)	Write a C program to calculate square root of a number using sqrt().
(13)	Write a C program to Use pow() to find power of a number.
(14)	Write a C program to Find the floor and ceiling value of a float using floor() and ceil().
(15)	Write a program to calculate sine, cosine, and tangent using sin(), cos(), atan().
(16)	Write a C program to Use fabs() to find the absolute value of a float number.
(17)	Write a C program to Generate exponential and logarithmic values using exp() and log().
(18)	Write a C program to Perform modulo operation on floats using fmod().
(19)	Write a C program to Check whether a character is digit, alphabet, etc. using isdigit(), isalpha(), etc.
(20)	Write a C program to Convert characters to upper and lower case using toupper() and tolower().
(21)	Write a C program to Convert string to float using atof().
(22)	Write a C program to Generate random numbers using rand() and srand().
(23)	Write a C program to Find absolute values using abs() and labs().
(24)	Write a C program to Use exit() function to terminate program.
(25)	Write a C program to Use strtoul() to convert string to unsigned long.
(26)	Write a C program to Free memory using free().
(27)	Write a C program to Allocate memory using malloc() and take input into array.
(28)	Write a C program to Use calloc() to allocate and initialize memory.
(29)	Write a C program to Use realloc() to resize memory block.
(30)	Write a C program to create a UDF to add two numbers.
(31)	Write a C program to create a UDF to calculate the square of a number.
(32)	Write a C program to create a UDF to calculate the cube of a number.
(33)	Write a C program to create a UDF to find the maximum of three numbers.
(34)	Write a C program to create a UDF to check whether a number is even or odd.
(35)	Write a C program to create a UDF to check whether a character is a vowel or consonant.
(36)	Write a C program to create a UDF to check whether a number is prime.
(37)	Write a C program to create a UDF to compare two strings.
(38)	Write a C program to create a UDF to find the maximum number in an array.
(39)	Write a C program to create a UDF to calculate area of circle, rectangle, and triangle using menu.
(40)	Write a C program to Demonstrate function call by value with a swap logic (values remain same).

(41)	Implement call by reference using pointers to swap two numbers.
(42)	Write a recursive function to calculate factorial of a number.
(43)	Write a recursive function to print Fibonacci series.
Structure	
(1)	Define a structure for a student (name, ID, marks), initialize its members, and print them using the dot operator.
(2)	Create a structure for an employee and accept data for one employee. Fields: emp_id, name, salary.
(3)	Create a union for student and accept data for one student Fields: rno, name
(4)	Create a structure for an employee and accept data for one employee. Fields: emp_id, name, salary.
Create udf to display data of structure.	
(5)	Store and display details of 5 students using array of structures. Fields: rollno, name
(6)	Create a program to store product information for N products using array of structures. Fields: product_id, name, price, quantity.
(7)	Store and display detail of student and display using pointer • Fields: roll, name
(8)	Write a program to dynamically allocate memory for a structure using pointers. Structure: employee with emp_id, name
(9)	Write a program to Find Size of a Structure
(10)	Write a program for nested structure
(11)	Program to demonstrate how only one member of a union can store value at a time.
(12)	Program to determine the size of a union and show memory usage.
(13)	Define a constant using macro and calculate area of square
(14)	Write a macro to find maximum of two numbers
(15)	Write a macro to find minimum of two numbers
(16)	Use #if, #elif, #else, #endif with macros
Pointer	
(1)	Write a program to declare a pointer, initialize it with the address of a variable and print the variable's value using the pointer.
(2)	Write a program to display the value and address of a variable using pointer.
(3)	Write a program to display sum of two numbers of using pointer
(4)	Write a program to swap the value of a variable using pointer.
(5)	Write a program to get maximum number from two numbers using pointer
(6)	Write a program to demonstrate the concept of pointer to pointer.
(7)	Write a program to access array elements using pointer notation.
(8)	Write a program to perform pointer arithmetic (increment/decrement) on an array and print the elements using a pointer.
(9)	Write a program to calculate the sum of array elements using pointer.
(10)	Implement a function that swaps two numbers using pointers passed as arguments.
(11)	Write a program to access array elements using a pointer and print the array.
(12)	Create a program where a function accepts a pointer as an argument to modify a variable's value (e.g., increment a number).
(13)	Write a program to allocate memory dynamically for an array using malloc, input values, and print them, then free the memory.

(14)	Implement a program to demonstrate the use of a pointer to a pointer to access and modify a variable's value.

BCA-1 CS-02: Networking & Internet Environment Minimum following exercise should be performed by the students during the semester	
1	Create a webpage profile.html using <html>, <head>, <title>, and <body>. Add your name, course, and college. Open it in a browser to view your personal homepage.
2	Build a digital resume using headings, paragraphs, line breaks, and formatting tags like <b>, <i>, <u>, etc. Include sections like Objective, Education, Skills, and Contact Info.
3	Simulate a short news article with <h2>, <p>, , <mark>, and tags. Include title, author, and content formatted to enhance readability.
4	Create a grocery list page using for priority items, for random items, and <dl> for descriptions. Use nested lists where needed.
5	Build a webpage with internal links to sections (About, Projects) and external links to sites like GitHub. Add mailto: and target="_blank" for better navigation.
6	Create a pricing page using special characters like €, ©, ≤, ≥, ™ inside a paragraph or table.
7	Use <img> to display a photo with alt, width, height. Use <audio> to embed a music file with controls and fallback text.
8	Design a table using <table>, <tr>, <th>, <td>. Include colspan, rowspan, and styling attributes. Add headings and rows for student data (Name, Subject, Marks).
9	Use <iframe> to create two sections: one showing a navigation menu (menu.html) and the other displaying content (content.html). Adjust width and height.
10	Create a form with fields: name, gender (radio), hobbies (checkbox), email, suggestions (textarea), and a submit button. Use appropriate input types and labels.
11	Design a layout using <header>, <nav>, <section>, <article>, <aside>, <footer>, <figure>. Add dummy content to mimic a news/blog layout.
12	Extend the previous form to use type="date", type="time", type="email", type="url", type="range", type="month", type="week", type="number" to collect more structured data.
13	Embed a video lecture using <video controls> and a music track using <audio controls>. Add autoplay and loop options for testing. Include fallback text.
14	Use <canvas> and JavaScript to draw a filled rectangle and a circle. Get context with getContext("2d"), then use fillRect() and arc() to draw shapes.
15	Create a homepage for a coffee shop using inline, internal, and external CSS. Compare results.
16	Design a portfolio page using inline styles (for one section), internal (for navigation), and external (for rest of the page).
17	Create a product grid with multiple product cards using class selector, and highlight

	one card using ID selector.
18	Build a blog list where links change on hover, first paragraph is styled uniquely, and alternate rows use :nth-child().
19	Create a typography demo page showcasing various font properties like size, weight, style, variant, and line height.
20	Design a news article layout with styled headings and paragraphs using alignment, spacing, and text-transform.
21	Build a travel website hero section with background image, repeat, position, size, and attachment settings.
22	Create a vertical sidebar navigation menu styled with different list-style types and custom images.
23	Design a multi-box layout and use different margins to explore element spacing and margin collapsing.
24	Add meaningful comments to a CSS file from a previous project explaining styling logic for each block.
25	Build a pricing table and apply various borders, styles, and border-radius to visually separate pricing plans.
26	Create a modern login page with linear and radial gradient backgrounds. Use background-blend-mode.
27	Create product cards with box-shadow, text-shadow, and transform effects like rotate, scale, and 3D tilt.
28	Create animated buttons that change size and color smoothly on hover using CSS transitions.
29	Build two similar boxes and use content-box and border-box to compare padding and layout behavior.
30	Create a modal popup or tooltip using position: absolute, relative, and fixed.
31	Design a responsive webpage with three breakpoints for desktop, tablet, and mobile. Change layout and visibility accordingly.
32	Build a responsive navigation bar and dashboard layout using all major flex properties including gap, justify-content, and align-items.
33	Design an interactive image gallery using overflow, text-overflow, filter, visibility, cursor, object-fit, and backdrop-filter.
34	Design a personal blog using Google Fonts for headings and custom fonts via @font-face for paragraphs.
35	Refactor an old project using BEM structure, e.g., .card__image--rounded, .button__icon--large. Apply consistent naming throughout.
36	Create a personal portfolio webpage that uses JavaScript to display a welcome message with your name and the current time.
37	Build a simple budget calculator where the user inputs income and expenses, and JavaScript variables store the values and display remaining balance.
38	Create a price calculator for an online shop that applies discounts, taxes, and calculates the final amount using arithmetic, logical, and assignment operators.
39	Design a student grading system where user inputs marks, and JavaScript displays grades based on conditions (A, B, C, Fail etc.).
40	Create a multiplication table generator where the user inputs a number and a loop displays its table. Use break to stop at a condition (e.g., stop at 50) and continue to skip a certain value.
41	Develop a feedback form that uses alert, confirm, and prompt to collect and confirm user data before submission.

42	Build a to-do list manager using arrays: add tasks, remove tasks, and show all tasks dynamically on the web page.
43	Write a program where clicking a button calculates and displays BMI (Body Mass Index) using a user-defined function.
44	Create a mini dashboard showing: current date/time (Date), capitalized user name (String), random daily quote (Array + Math.random()), and an age calculator (Math.floor).
45	Create a responsive card: - Change background color onmouseover - Zoom in image ondblclick - Show keycode on onkeypress - Auto-capitalize name field onblur - Display message onload - Reset form onreset and more.
46	Build a navigation system: Use DOM to dynamically change content on link click. Add buttons for Back and Forward using history.back() and history.forward().
47	Create a registration form that validates required fields, email format, password match, and phone number using JavaScript before allowing submission.
48	Create a personal portfolio webpage using Bootstrap CDN. Include your photo, bio, education, and social media links.
49	Design a responsive product listing page for an e-commerce store showing 6 products in a grid. Use containers, rows, columns, offset and reordering to make it look good on mobile, tablet, and desktop views.
50	Create a blog post layout using headings, paragraphs, blockquotes, and text utilities. Include at least one motivational quote styled distinctively.
51	Build an attendance record table for students with striped rows, hover effects, and responsive design. Add a heading and caption.
52	Display a gallery of 4–6 images with rounded corners, thumbnails, and responsive classes. Add captions below each image.
53	Create a student admission form using Bootstrap form classes. Include inputs for name, email, gender (radio), skills (checkbox), course (dropdown), and submit button.
54	Design a responsive navbar for a college website with logo, navigation links (Home, Courses, Contact), and a search bar on the right.
55	Create a tabbed section to show semester-wise syllabus. Each tab shows syllabus for one semester using lists or cards.
56	Add a dropdown menu under the “Courses” section in the navbar to list BCA, BBA, BSc. Show a different layout for each on selection.
57	Design a control panel with grouped buttons like “Add Student”, “Edit Record”, “Delete Record” using button groups and contextual colors.
58	Add breadcrumb navigation to a student dashboard that shows: Home > Dashboard > Profile. Make it responsive.
59	Create a paginated list of registered users or student records showing 5 entries per page. Use Bootstrap pagination.
60	Add success, warning, and error alerts to a registration form for different submission states. Use badges to show notifications or counts.
61	Simulate a course completion status bar for 5 different courses. Each course should have a labeled progress bar showing completion percentage.
62	Create an FAQ section with accordion where each panel opens to show answer when clicked. Use Bootstrap collapse/accordion component.
63	Build a staff directory using Bootstrap cards. Each card should display staff photo, name, designation, and contact details.
64	Add a "Contact Us" modal popup on a college website. Trigger it from a “Get in

	Touch" button. Include name, email, message, and close buttons.
65	Create a section with 4 colored info boxes (e.g., Notices, Events, Exams, News) using background and text color utilities.
66	Style a set of cards with different border styles and use display utilities to toggle visibility of a "More Info" section.
67	Create a fixed top notification bar and a scrollable list of items within a card using overflow and position utilities.
68	Add margin and padding to various sections of a webpage to ensure proper layout and spacing between elements. Use spacing classes only.
69	Design a testimonial section where user images and text are vertically aligned. Use text utilities to center and format content properly.

	BCA-2 **CS-08: Data Structure Using C Language** **Minimum following exercise should be performed by the students during the semester**
1.	Write a C program to count the number of steps taken to perform linear search.
2.	Compare time complexity of linear vs binary search using a program.
3.	Show space used by dynamic array vs static array.
4.	Write a function with O(n^2) time complexity and print execution count.
5.	Illustrate a function with both best-case and worst-case scenarios.
6.	Write a C program to create a text file named student.txt. The program should ask the user to enter a student's name and marks, then write that data into the file using fprintf(). Finally, close the file using fclose().
7.	Write a C program to read the student details (name and marks) stored in the student.txt file using fscanf() and display them on the screen.
8.	Write a C program to open a file named sample.txt, move the file pointer to the 5th byte using fseek(), and display the file pointer position using ftell(). Also, read the character at that position.
9.	Write a C program that performs the following tasks step-by-step.
a.	Create a binary file numbers.dat and write 5 integers into it.
b.	Read and display the integers from the file and detect the end of the file.
c.	Move the file pointer to the end of the file and display its position.
d.	Reset the file pointer to the beginning of the file.
e.	Create a log file log.txt to record the total number of integers.
f.	Read and display the contents of the log file.
g.	Redirect the input stream to read from the log file.
h.	Check if any file error occurred during file operations.
i.	Rename the binary file to backup.dat.
j.	Delete the log.txt file.
10.	Write a C program that accepts details of n students (roll number, name, and marks), stores them in a file, and then reads and displays the records from the file.
11.	Write a C program that accepts two integers from the command line and displays their sum.
12.	Write a C program that takes a filename as a command line argument and displays the total number of characters in that file.
13.	Write a C program that accepts multiple command line arguments and displays the length of each argument.
14.	Write a C program that accepts a number as a command line argument and checks whether the number is prime or not.
15.	Write a C program that accepts any number of command line arguments and displays how many were provided (excluding the program name).
16.	Write a C program to sort an array of integers using the Bubble Sort technique.
17.	Write a C program to sort an array of strings using Bubble Sort.
18.	Write a C program to sort an integer array using Insertion Sort.
19.	Sort an array of strings using Insertion Sort in C.
20.	Write a C program to implement Quick Sort on integers.
21.	Write a C program to implement Quick Sort on strings.
22.	Write a C program to sort positive integers using Bucket Sort.
23.	Write a C program to sort integer arrays using Merge Sort.
24.	Write a C program to sort integers using Selection Sort.

25.	Write a C program to sort integers using Shell Sort.
26.	Write a C program to implement Index Searching on a sorted integer array using a simple index table.
27.	Write a C program to perform Index Searching on a sorted list of strings using index positions.
28.	Write a C program to search an integer element using Sequential Search.
29.	Write a C program to perform Linear Search to find a string in a list of strings.
30.	Write a C program to implement Binary Search on a sorted integer array.
31.	Write a C program to implement Binary Search on a sorted string array.
32.	Write a menu-driven C program to perform various stack operations using arrays: Push, Pop, Peek, Display. Demonstrate stack overflow and underflow conditions also.
33.	Write a menu-driven C program to implement queue operations (enqueue, dequeue, peek, display) using arrays. Demonstrate queue overflow and underflow conditions also. Check whether the queue is full or empty also.
34.	Write a C program to implement a circular queue using arrays.
35.	Write a C program to implement a double-ended queue (deque) using arrays.
36.	Write a C program to implement a priority queue using arrays.
37.	Write a C program to create and display a singly linked list.
38.	Write a C program to insert a node at the beginning of a singly linked list.
39.	Write a C program to insert a node at the end of a singly linked list.
40.	Write a C program to insert a node after a specified node in a singly linked list.
41.	Write a C program to traverse and display all nodes of a singly linked list.
42.	Write a C program to delete a node from a singly linked list.
43.	Write a C program to update the data of a specific node in a singly linked list.
44.	Write a C program to merge two singly linked lists into a single linked list.
45.	Write a C program to reverse a singly linked list.
46.	Write a C program to create and display a doubly linked list.
47.	Write a C program to insert a node at the beginning of a doubly linked list.
48.	Write a C program to insert a node at the end of a doubly linked list.
49.	Write a C program to insert a node after a specified node in a doubly linked list.
50.	Write a C program to traverse and display all nodes of a doubly linked list in both forward and reverse order.
51.	Write a C program to delete a node from a doubly linked list.
52.	Write a C program to update the data of a specific node in a doubly linked list.
53.	Write a C program to create and display a circular linked list.
54.	Write a C program to insert a node at the beginning of a circular singly linked list.
55.	Write a C program to insert a node at the end of a circular singly linked list.
56.	Write a C program to insert a node after a specified node in a circular singly linked list.
57.	Write a C program to traverse and display all nodes of a circular singly linked list.
58.	Write a C program to delete a node from a circular singly linked list.
59.	Write a C program to update the data of a specific node in a circular singly linked list.
60.	Write a C program to create and display a header linked list.
61.	Write a C program to merge two singly linked lists into a single linked list.
62.	Write a C program to reverse a singly linked list.
63.	Write a C program to implement a binary tree and perform basic node insertion
64.	Write a C program to perform Inorder traversal of a binary tree.
65.	Write a C program to perform Preorder traversal of a binary tree.
66.	Write a C program to perform Postorder traversal of a binary tree.

67.	Write a practical C program to perform DFS traversal on a graph and display the traversal order.
68.	Write a practical C program to perform BFS traversal on a graph and display the traversal order.

BCA-2 **CS-09 Web Programming** **Minimum following exercise should be performed by the students during the semester**	
PHP Practicals	
(1)	Create a simple PHP page to display "Hello, World!"
(2)	Display current date and time using PHP date() function
(3)	Create a calculator using HTML and PHP
(4)	Form handling: Accept name, email, and display it using PHP
(5)	Validate a form (email, name) with PHP
(6)	Create a login system using PHP (without database)
(7)	Use sessions to store and display user information
(8)	Use cookies to store a username and greet user on reload
(9)	Create a PHP script to upload files to the server
(10)	Create and use user-defined PHP functions
(11)	Demonstrate string functions in PHP (like strlen(), str_replace())
(12)	Create an associative array and loop through it
(13)	Read and write files using fopen(), fwrite()
(14)	Use include and require statements for modular PHP
(15)	Use isset() and empty() to check variables
(16)	Create a simple contact form and send data to email
(17)	Build a feedback form and store data in a text file
(18)	20. Create a dynamic menu with PHP arrays
AJAX Practicals	
(1)	Load data from server without refreshing the page
(2)	Submit a form using AJAX and display result
(3)	Validate email availability using AJAX (with dummy PHP)
(4)	Create live search using AJAX
(5)	Load dropdown options based on selection (e.g., country → state)
(6)	Fetch data from a JSON file using AJAX
(7)	Load external content (like text file) into a div
(8)	Create an autocomplete textbox using AJAX
(9)	AJAX polling: show server time every few seconds
(10)	Submit login form via AJAX and show success/error
(11)	Use AJAX to add items to a cart without refreshing
(12)	Display database records using AJAX and PHP
(13)	Update part of a page using XMLHttpRequest
(14)	Use jQuery $.ajax() method to fetch PHP output
(15)	AJAX-based feedback form with real-time validation
(16)	Create a to-do list using AJAX and PHP
(17)	AJAX-based image upload preview
(18)	Show loader while AJAX call is in progress
(19)	Handle AJAX errors (404, timeout)
(20)	AJAX with XML: Load and display XML data
MySQL Practicals	
(1)	Create a MySQL database and table using PHPMyAdmin
(2)	Insert, update, delete records using SQL queries
(3)	Connect PHP to MySQL database
(4)	Display all records from a table using PHP

(5)	Create a login system using MySQL and PHP
(6)	Search records in a table from user input
(7)	Sort records in ascending/descending order
(8)	Display limited records with pagination
(9)	Use aggregate functions (COUNT(), SUM())
(10)	Prevent SQL Injection using prepared statements
(11)	Build a simple blog with title, content, and MySQL
(12)	Implement a registration form with PHP + MySQL
(13)	Create a product listing with category filter
(14)	Create a MySQL backup and import it
(15)	Display top 5 recent posts using SQL LIMIT
(16)	Use PHP to update records via MySQL
	jQuery Practicals
(1)	Change content and style of an element on click
(2)	Create a hide/show toggle button
(3)	Fade in/out effect on image or text
(4)	Slide up/down menu effect
(5)	Form validation using jQuery (required fields)
(6)	Animate an element's position or size
(7)	Create tabs using jQuery
(8)	jQuery event handling (click, dblclick, keyup)
(9)	Load content via AJAX using jQuery load()
(10)	Create a dropdown dependent menu (city based on state)
(11)	jQuery image slider
(12)	jQuery modal popup
(13)	Toggle class on button click
(14)	Scroll to top button using jQuery
(15)	jQuery accordion
(16)	Create a to-do list with jQuery
(17)	Drag and drop items using jQuery UI
(18)	jQuery hover effect for image zoom
(19)	jQuery input character counter
(20)	jQuery autocomplete using AJAX and PHP
	OOP in PHP Practicals
(1)	Create a simple PHP class and object – Define a class `Car`, create object and call a method.
(2)	Define class with constructor and destructor – Show how constructor initializes object and destructor cleans up.
(3)	Use of access modifiers (public, private, protected) – Create class members with different visibility and access them.
(4)	Use of getter and setter methods – Implement encapsulation using `getName()` and `setName()`.
(5)	Create a class with multiple methods – E.g., `BankAccount` with deposit, withdraw, checkBalance.
(6)	Demonstrate inheritance with parent and child classes – Class `Animal` with subclass `Dog` overriding a method.
(7)	Use `final` keyword to prevent method override – Create a final method in a base class.

(8)	Create and use abstract classes – Abstract class `Shape`, derived classes `Circle`, `Rectangle`.
(9)	Use interfaces to define behavior – Interface `Movable`, implemented by class `Car` and `Bike`.
(10)	Demonstrate constructor overloading using default parameters – Create flexible constructors in a class.
(11)	Create a static method and property – Show how to use class-wide data like `count` of instances.
(12)	Demonstrate method overriding and `parent::` keyword – Call parent class method from child using `parent::`.
(13)	Implement traits to share functionality across classes – Use PHP traits to add common methods in multiple classes.
(14)	Use namespaces to organize code – Create two classes with the same name in different namespaces.
(15)	Create a class with constants and use them – E.g., `const STATUS_ACTIVE = 1;`.
(16)	Implement dependency injection in PHP classes – Pass database or logger object through constructor.
(17)	Create a simple autoloader using `spl_autoload_register()` – Auto-include classes from folder structure.
(18)	Create and handle exceptions using try-catch block – Throw custom exceptions in classes like `InvalidAgeException`.
(19)	Implement CRUD operations with OOP – Create `User` class with `addUser()`, `deleteUser()`, etc., using MySQL.
(20)	Create a complete mini project using OOP – e.g., Library Management or Student Management System using OOP, MySQL, and PHP.