GNSU Handbook Of BCA (Bachelor of Computer Application) 3-Years Programme



AY-2023-24

August-2023

Table of Contents

S. No.	Title	Page
1	1.1 Teaching Learning Process in Faculty of Information Technology	3-4
	1.2 Flowchart of Teaching Learning Process in Faculty of Information	
	Technology	
2	Admission Guidelines	
3	B.C.A.(3 Years-6 Semester) guidelines	4-5
4	Industry Ready courses in curriculum- IT Workshop and Skilling	5
5	General Course Structure & Theme-	
	5.1 Definition of Credit	5-6
	5.2 Range of Credit	
	5.3 Course level coding scheme	
	5.4 Mandatory Induction Program in 1 st Sem BCA	
6	Semester Wise Structure-B.C.A.(3 Years-6 Semester) Courses	
	* details in Appendix-1	7-9
7	Mandatory Non-Credit Courses	
8	Evaluation Scheme and Rubrics	
	a. Rubrics for Theory Courses	
	b. Rubrics for Practical Courses	
	c. Rubrics for Summer Internship/Projects/Seminar etc.	9-12
	d. Mapping of Marks to Grades	
	e. Promotional Policy	
	f. Calculation of DGPA, CGPA for one, two, three and four-year programmes.	
9	Question paper format for In-Sem and End Sem Examinations	12-14
10	Appendix 1: Detailed Syllabus Semester-I to Semester VIII	15



1.2 Teaching Learning Process- Flowchart



	The academic calendar e	vents and time table are shared with the students. The faculty prepares the
	lesson plan for the allotte	d subject. For better delivery of teaching learning process, the faculty prepares
	the course plan and learn	ing methodology for the allotted subject.
	The faculty conducts the	classes as per the lecture and course plan. Online feedback from the students
	are collected and analyse	d by the Departmental Academic committee (DAC) . If any inconsistencies are
	found by DAC in the teach	ing learning process, the concerned faculty are advised to modify the teaching
	methodology and also att	end Faculty development programmes.
	Two internal assessments	s are conducted for a particular subject per semester. The attainment of COs
	are analysed for taking	remedial actions. Remedial actions include identification of slow, and fast
	learners.	
	For the slow learners (≤4	5% marks in class test) the following methods of improvement are adopted-
	Peer teaching, Counsellin	g, mentoring, intimation to parents and conducting extra classes.
	Fast learners (>75% mark	ts in class test) are motivated to continue the achieve excellency and they are
	encouraged to participate	e in co-curricular and extra-curricular activities.
	A candidate shall be eligible of Computer Applications)	e for appearing in the Entrance Test leading to admission to BCA(Bachelor Semester I, if he/she has passed the following:
	Admission to 3-Year	45% marks or equivalent grade in "10+2" examination (any discipline) from any recognized board preferably Mathematics/ any subject in Computer Science. OR
	BCA	A pass in diploma in Commercial Practice or equivalent (The University will offer suitable bridge courses such as Mathematics etc., forthe students coming from diverse backgrounds to prepare Level playing field and desired learning outcomes of the programme)
	** Presently BCA-3 year Pr	ogramme is running in Faculty of Information Technology-Sem-1 to Sem-6 is
	applicable	
3	B.C.A. (3 Years-6 Semester) guidelines:
	a. Admission in BCA I- Test.	Semester programme of study shall be made on merit in the Entrance
	b. The Faculty shall constituted under th two senior-most me accordance with the	have an Admission Committee for BCA undergraduate admission, ne provisions of Ordinances and consisting of the Dean or his nominee and embers of the teaching staff of the Faculty. Admission shall be made in ese ordinances and the rules made there under.
	c. Admission cannot, h or re-admission of a	nowever, be claimed by any candidate as a matter of right. The admission candidate shall be entirely at the discretion of the Admission Committee

which may refuse to admit any student without assigning anyreason there for. d. On his/her selection for admission to the BCA Semester I programme, the candidate shall, within the time fixed by the Dean, Faculty of Information Technology deposit the fees prescribed for the programme. If the candidate fails to deposit fees within the stipulated time, his/her selection shall automatically stand cancelled. e. Reservation in admission will be made as Govt. of Bihar rules Intake Course Intake 120 BCA (3-years, 6 Semesters) **Documents Required** Scanned copies of academic details. Scanned passport size photograph of the candidate in JPG/JPEG format • Scanned clear signature of the candidate in JPG/JPEG format • A valid e-mail ID. Scanned copies of Category certificate, PwD certificate, if applicable. • Migration certificate • • Transfer certificate Character certificate • **Other documents if any • Academic Calendar - Each academic session is divided into two semesters of approximately an Odd semester (July- December) and Even semester (January-May). 4. Industry Ready courses in curriculum- IT Workshop and Skilling The courses which make the students Industry ready has been incorporated in regular BTECH CSE curriculum. IT Workshop & Skilling-1 which may lead to Oracle or Microsoft Certifications has been incorporated in 4th Semester of three year BCA curriculum. IT Workshop & Skilling-2 which may lead to Hadoop certifications has been incorporated in 7^h Semester of four year BCA curriculum. These courses will be conducted during Summer breaks. Skilled instructors from Industry will be conducting these courses to make the students Industry Ready. After the completion of the courses the students may be motivated to appear for different certification programs by Oracle, Microsoft etc which will ultimately lead to better placements. The skills which are required by the industry need to be updated as per requirements in IT Workshop & Skilling-1 and 2. **Tentative Skills that may be required by the Industry** – IT Java Script, Java, Rust, ReactJS, NumPy and Pandas, Kotlin, Django, Angular JS Front End Skills, Java Full Stack, Python, Snowflake, Cybersecurity, Cloud Native development such as Kubernetes, Microservices, Blockchain development, Docker and Containerisation, Azure ML, RPA, Edge Computing, Data Architect, Data Scientist and Analytics skills.

5.	General Course Structure & Theme-
	5.1 Definition of Credit:
	1 Hr. Lecture (L) per week1 Credit1 Hr. Tutorial (T) per week1 Credit1 Hr. Practical (P) per week0.5 Credit2 Hours Practical (P) per week1 Credit
	5.2. Range of Credits: Credit Requirements: The candidates shall complete courses equivalent to a minimum of
	 140 credits to become eligible for the Regular Bachelor in Computer Applications Degree(3-Years,6 semesters) 180 credits to become suitable for the Bachelor in Computer Applications Degree with Honours (4-years,8 semesters). An individual seeking to continue to the Bachelor in Computer Applications (Honours) would normally have completed all requirements of the 3-year Bachelor in Computer Applications Degree.
	5.3 Course level coding scheme: Three-digit number (odd numbers are for the odd semester courses and even numbers are for even semester courses) used as suffix with the Course Code for identifying the level of the course. Digit at hundred's place signifies the year in which course is offered.
	e.g. 101, 102 etc. for first year. 201, 202 Etc. for second year.301, 302 for third year.
	Sessional (Project, Seminar, etc) e.g. PCC-CS-681i.e 6-Sixth Sem,8-Sessional, 1- sequence.
	Laboratory- PCC-CS-691i.e 6-Sixth Sem,9-Laboratory, 1- sequence Total credit for BCA (3 Years) to be awarded- 140 Total credit for BCA (4 Years) to be awarded- 180

5.4 Mandatory Induction Program: The Essence and Details of Induction program can also be understood from the 'Detailed Guide on Student Induction program', as available on AICTE Portal, (Link: <u>https://www.aicteindia.org/sites/default/files/Detailed%20Guide%20on%20Student</u>

%20Induction%20program.pdf).

Induction program (mandatory)	Three-week duration
Induction program for students to be offered right at the start ofthe first year.	 Physical activity Creative Arts Universal Human Values Literary Proficiency Modules Lectures by Eminent People Visits to local Areas Familiarization to Dept./Branch 8 Innovations

~
h
v

Semester-wise subject structure of B.C.A (3 years-6 semesters) -

		1st Semester-Theory			•		Credit
1	BC23-101	Digital Electronics	3	1	0	4	
2	BC23-102	Programming using C	3	1	0	4	
4	BC23-103	Discrete Mathematics	3	1	0	4	
5	BC23-104	PC Software	3	1	0	4	20
		1st Semester-Laboratory					
6	BC23-192	Programming using C(Lab)	0	0	4	2	
7	BC23-194	PC Software Laboratory	0	0	4	2	
		2nd Semester-Theory					
1	BC23-201	Computer Architecture & Organization	3	1	0	4	
3	BC23-202	Data Structures using C	3	1	0	4	
4	BC23-203	Mathematics-I	3	1	0	4	
5	BC23-204	English Language and Communication	3	1	0	4	20
		2nd Semester-Laboratory					
6	BC23-292	Data Structures using C Laboratory	0	0	4	2	
٦	DC22 204	Business Presentation and Language	_	0	4	2	
/	BC23-294	Laboratory	0	0	4	2	
		3rd Semester Theory	•	•	-		
1	BC23-301	Software Engineering	3	1	0	4	
2	BC23-302	Operating Systems	3	1	0	4	
3	BC23-303	Computer Graphics and Internet	3	1	0	4	
4	BC23-304	OOPs using C++	3	1	0	4	
5	BC23-305	Management and Accounting	3	1	0	4	26
		3rd Semester Laboratory					26
6		Operating Systems & Shell programming	0	0	4	2	
0	BC23-392	Laboratory (Linux to be included)	0	U	4	Z	
7	BC23-393	Computer Graphics and Internet Laboratory	0	0	4	2	
8	BC23-394	OOPs using C++ Laboratory	0	0	4	2	
		4th Semester Theory					
1	BC23-401	Database Management Systems	3	1	0	4	
2	BC23-402	Numerical Methods	3	1	0	4	
3	BC23-403	Web Technology	3	1	0	4	
4	BC23-404	Values and Ethics in Profession	3	0	0	3	
		4th Semester Laboratory					27
5	BC23-491	Database Management Systems laboratory	0	0	4	2	
6	BC23-492	Numerical Methods using C Laboratory	0	0	4	2	
7	BC23-493	Web Technology Laboratory	0	0	4	2	
8	BC23-494	IT Workshop & Skilling-1	0	0	12	6]
		5th Semester Theory			-	-	
1	BC23-501	Computer Networks	3	1	0	4	
2	BC23-502	Python Programming	3	1	0	4	
3	BC23-503	Design and Analysis of Algorithms	3	1	0	4	
4	BC23-		2	4	_	4	27
4	E501A/B/C	Elective-1	3	L T		4	27
		5th Semester Laboratory	<u> </u>				
5	BC23-591	Computer Networks laboratory	0	0	4	2	1
-							

7	BC23-581	Minor Project-1	0	0	6	3	
8	BC23-582	Industrial Training/Internship (6 to 8 Weeks)	0	0	0	4	
		6th Semester Theory					
1	BC23-601	Environment and Ecology	3	1	0	4	
2	BC23-E601 A/B/C	MOOCs-Open Electives preferably be opted from the NPTEL/SWAYAM Platform*.	3	1	0	4	
3	BC23-E602 A/B/C	MOOCs-Open Electives preferably be opted from the NPTEL/SWAYAM Platform*.	3	1	0	4	2'
		· · · · · · · · · · · · · · · · · · ·					
		6th Semester Laboratory			r		
4	BC23-681	Major Project-1	0	0	12	6	
5	BC23-682	Seminar and Group Discussion	0	0	4	2	
		List of Electives in the 5th and 6th Sem BCA					
1	BC23-E501A	Image Processing	3	1	0	4	
2	BC23-E501B	Pattern Recognition	3	1	0	4	
З	BC23-E501C	Compiler Design	3	1	0	4	
4	BC23- E601A/B/C	the semester e.g. Cryptography and Network Security, Web and Social Media Analytics, Data Mining	4	0	0	4	
7	BC23- E602A/B/C	MOOCs basket will be declared at the start of the semester e.g. E-Commerce and ERP, Introduction to Digital Marketing, Business Communication & Value Science	4	0	0	4	
		Total credit for BCA 3 years to be awarded= 140)				
	7. Mandatory	Course-Non Credit					
1	MC-CS-401	Environmental Sciences	2	0	0	Non-C	redit
1	MC-CS-501	Constitution of India/ Essence of Indian Knowledge & Tradition	1	0	0	Non-Credit	
*M Info	OOCs Regarding: W ormation Technolog While opting for a d	IOOCs basket will be declared at the beginning of the se gy based on emerging technologies. The courses has to l course for pursuing the Open Elective, a student needs t	mest be ch	er by osen f sure th	Dean From t	Faculty o he baske	f t.
i) Tl ii) T iii) I Yea	he duration of the o The course must no Date of Exam and p or BCA) i.e. January	course must minimum of 12-Weeks. t be covered in previous semesters of the program. ublication of result should be within the tenure of the B to June of every Year.	CA 6	th Sem	ester	(8 th Sem-	Four
	The student should	register and submit the details of the source at the tim	o of c		oont o	fetheom	

8.	Evaluation Scheme:
	It is advised to refer page numbers 48-51 of the document "Examination Reform Policy November 2018" published by AICTE to get some examples of assessment rubrics (https://www.aicte-india.org/sites/default/files/ExaminationReforms.pdf). There are several good examples of assessment rubrics available on the web when searching with the string "examples of assessment rubrics". There are many good videos available on YouTube on examples and strategies for preparing assessment rubrics.
	Since assessment rubrics are not generic in nature and depend on courses, topics,

assessment strategies of Information Technology Sessional evaluation.	of individual faculties will follow the following	etc., it is suggested that Fac ng rubrics for Theory, Laborat	culty of ory and
 a. Rubrics for Theory C The internal assessment 70 marks. For passing th 100 (Internal assessment) 	Courses: will be of 30 marks and the subject the students t and end semester asse	l end semester examinations w should obtain at least 40 mark essment).	ill be of s out of
Suggestive rubrics for In be Average of Best of th	ternal Assessment : Calor ree CA plus 5 marks of A	culation of Internal assessment Attendance.	number will
Continu	ous Assessment	Activities	
	CA1	Quiz/ Assignment	
	CA2	Internal test (Pen and pa	per)
	CA3	PPT presentation/ Group Dis	cussion
	CA4	Internal test(Pen and pa	per)
Sessional Exams- (Proj semester examinations should obtain at least 50 of report, performance based on course outcom external examiners as pe c. Mapping of Marks to Grades Each course (Theory/Practical)is and the mapping of marks to gra	ect, Seminar, Group E will be held for 100 ma marks out of 100 . Eval in viva-voice, presentati es. The Project Viva and er rules of the University to be assigned 100 mar ades may be done as per	Discussion, Internship, Trainin arks. For passing the subject t luation will be based on work o ion etc. for which rubrics may practical examinations will pre /. ks, irrespective of the number r the following table:	g etc.) end he students lone, quality be designed ferably have of credits,
Classification	Letter Grade	Score on 100 percentage point	Points
Outstanding	0	100 to 90	10
Excellent	E	89 to 80	9
Very Good	A	79 to 70	8
Good	В	69 to 60	7
Fair	C	59 to 50	6
Below Average	D	49 to 40	5
Failed	F	Below 40	2
Incomplete	1		2
	-		

d. Promotional Policy: Candidates will be eligible for promotion to the next semester without clearing all end semester theory courses of earlier semesters if,

- a. Candidate has minimum attendance percentage of 75% in the previous semester
- b. Candidates must have appeared for all internal examinations and has secured marks in Continuous Assessments, Sessional Examinations, Practical Examinations
- c. Candidates must have applied for appearing in the end semester examinations and have valid admit card in previous semester

Candidates failed to achieve the minimum benchmarks as mentioned in (a), (b), (c) for promotion will not be eligible for promotion to the next higher semester.

Candidates will appear in the end semester theory examinations as back log candidate in corresponding semester, of subsequent academic year. Marks scored in Continuous Assessments, Sessional Examinations, Practical Examinations during attending regular semester with minimum qualifying attendance would be carried all through. Backlog candidates would be allowed to appear in the end semester examinations only to achieve qualifying marks of the paper concerned. For 7th and 8thSemester backlog students, backlog examinations may be conducted after two months from the date of result publication.

If any candidate fails to secure minimum qualifying marks (pass marks) in sessional or practical examinations would suffer year lag and they have to continue the semester concerned afresh in the next academic year. In the internal examination of 30 marks -Cumulative Assessment 25 plus 5 marks for attendance.

The marks of a back log paper will be determined from the marks obtained in theory examination and marks of the continuous evaluation of the regular semester. No up-gradation of internal/continuous assessment marks would be allowed.

If any candidates fail to achieve any of the three conditions above (a, b & c) in any semester (say, 1st semester), they would not be allowed to continue their study in the next semester (i.e., 2nd semester) and they have to fulfill the academic regulations by enrolling them in the next academic year from the discontinued semester (i.e. 1st semester) and so on.

However, there would not be any limit of number of back papers to continue their study in subsequent semester as regular candidate.

e. Calculation of DGPA, CGPA for one, two, three and four year programmes.

- Result Status: X=Not eligible for Semester Promotion/Degree; XP=Eligible for Promotion with Backlogs; P=Passed and Promoted.
- The method of calculation of Grade Point Average is as follows
- SGPA (Semester Grade Point Average) = $\frac{Credit Index}{\sum Credits}$
- YGPA (Yearly Grade Point Average) = $\frac{\text{Credit Index Odd Semester + Credit Index Even Semester}}{\sum \text{Credits Odd Semester + D.C. We The set of the set$
- For final Degree Grade Point Average (DGPA) the calculation is as under

DGPA = $\frac{\text{YGPA 1 + YGPA2 + 1.5* YGPA3 + 1.5* YGPA4}}{5}$

(For 4 Year Course)		
DGPA	=	YGPA2 + 1.5* YGPA3 + 1.5* YGPA4 4
(For Lateral Entry Studer	nts)	
DGPA	=	YGPA 1 + YGPA2 + YGPA3 3
(For 3 Year Course)		
DGPA	=	YGPA 1 + YGPA2 2
(For 2 Year Course)		
DGPA	=	YGPA1
(For 1 Year Course)		
CGPA = $\frac{\sum_{k=1}^{K=n} Credit Index}{\sum_{k=1}^{k=n} Credit of}$ Where n = 4 for 2 years n=6 for 3 years p n = 8 for 4 years n = 10 for 5 years	DINT AVAR <u>a of kth sem</u> k th Semesta programme programme s programm	AGE (CGPA)

Stu	dent Roll No	. Studen	it Name:
	A	Academic year:	2023-24
	GNSU	B.C.A ,2023	nations-I Batch
	Subject Code:	1 st Semes	Subject Name
Гiп	ie: 1 hours		Max. Marks
This for cond	template is recommended for co courses with five COs. The COs lucted.	arses with 4 COs and two can be altered in the te	o In-Sem Tests. Can also be fo emplate depending on the test
	(Assume any missing data su	itably and design adeq	quate hypothesis, if required
		Part-A	(3X 1M=3M)
	Answer	Any Three Question	ns m the May PTL of (T)
	Q. No. 3, 4 from CO1 Pres	arred to be at lower BTL th	an the Max B1L of CO1
1.	Q. NO 5, 4 Hom CO2 Field	ited to be at lower DTD an	
2.			
з.			
4.			
		Part-B	(2 X 4M=8M)
	Answ	er Any Two Question	5
<u>0</u>	No. 5, 0 from COI Preferred to be at	lower BIL than the Max B	SIL of COP
5.		one bib and are may b	12 11 001
-			
- 0 . 7			
8.			
		Part-C	(2 X 7M=14M)
	An	swer ALL Questions	
	Q. No. 9,10 from COI and ha	ive a internal choice betw	een Q.No.9 and Q.No.10
	O No II Diffrom COV and by		een Q.No.II and Q.No.I2
	Q. 140. 11,12 Hom CO2 and h	we a internal choice betw	
9.	Q. No. 9 from COI Preferred to b question must be of Max. BTL of t	e at lower BTL than the N he CO1, may 2 sub questio (Or)	fax BTL of CO1, At least one su ms.
9. 10.	Q. No. 9 from COI Preferred to b question must be of Max. BTL of t Q. No.10 from COI Preferred to b question must be of Max. BTL of t	e at lower BTL than the N he CO1, max 2 sub questio (Or) e at lower BTL than the N he CO1, max 2 sub questio	fax BTL of CO1, At least one su ms. Jax BTL of CO1, At least one su ms.
9. 10. 11.	Q. No. 9 from COI Preferred to b question must be of Max. BTL of 1 Q. No.10 from COI Preferred to b question must be of Max. BTL of t Q. No.11 from CO2 Preferred to b question must be of Max. BTL of t	e at lower BTL than the N he CO1, max 2 sub questio (Or) we at lower BTL than the N he CO1, max 2 sub questio we at lower BTL than the N he CO1, max 2 sub questio	fax BTL of CO1, At least one su ms. Max BTL of CO1, At least one su ms. Max BTL of CO2, At least one su ms.

	GNSU	Academic year: 2 Internal Examin B.C.A, 2023 E 1** Semest	2023-24 ations-II latch er
S	ubject Code:		Subject Name
Tin	ne: 1 hours		Max. Marks:
Thi for con	s template is recommended for courses with five COs. The C ducted.	courses with 4 COs and two Is Os can be altered in the tem	n-Sem Tests. Can also be follo plate depending on the test b
	(Assume any missing data	suitably and design adeque	te hypothesis, if required)
		Part-A	(3X 1M=3M)
	Anst	wer Any Three Questions	
	Q. No. 1, 2 from CO3 P	reterred to be at lower BTL than	the Max BTL of CO3
L	Q. No 3, 4 from CO4 P	referred to be at lower BTL than	the Max BTL of CO4
1.			
<u>2</u> .			
4.			
<u> </u>		Part-B	(2 X 4M-8M)
	Ал	swer Any Two Ouestions	(2 11 11-011)
0.	No. 5, 6 from CO3 Preferred to be	e at lower BTL than the Max BTL	, of CO3
Q.	No 7, 8 from CO4 Preferred to be	at lower BTL than the Max BTL	of CO4
5.			
-			
7.			
8.			
		Part-C	(2 X 7M=14M)
		Answer ALL Ouestions	(,
I	Q. No. 9,10 from CO3 and	i have a internal choice between	1 Q.No.9 and Q.No.10
\vdash		i have a mternal choice between	n Q.No.11 and Q.No.12
	Q. No. 11,12 from CO4 and	o be at lower BTL than the Max	BTL of CO3, At least one sub
9.	Q. No. 11,12 from CO4 and Q. No. 9 from CO3 Preferred t question must be of Max. BTL	of the CO3, max 2 sub questions. (Or)	
9. 10.	Q. No. 11,12 from CO4 and Q. No. 9 from CO3 Preferred t question must be of Max. BTL Q. No.10 from CO3 Preferred t question must be of Max. BTL	of the CO3, max 2 sub questions. (Or) to be at lower B1L than the Max of the CO3, max 2 sub questions.	BTL of CO3, At least one sub
9. 10. 11.	Q. No. 11,12 from CO4 and Q. No. 9 from CO3 Preferred t question must be of Max. BTL Q. No.10 from CO3 Preferred t question must be of Max. BTL Q. No.11 from CO4 Preferred t question must be of Max. BTL	of the CO3, max 2 sub questions. (Or) to be at lower BTL than the Max of the CO3, max 2 sub questions. to be at lower BTL than the Max of the CO4, max 2 sub questions.	BTL of CO3, At least one sub BTL of CO4, At least one sub
9. 10. 11.	Q. No. 11,12 from CO4 and Q. No. 9 from CO3 Preferred t question must be of Max. BTL Q. No.10 from CO3 Preferred t question must be of Max. BTL Q. No.11 from CO4 Preferred t question must be of Max. BTL	of the CO3, max 2 sub questions. (Or) to be at lower B1L than the Max of the CO3, max 2 sub questions. to be at lower B1L than the Max of the CO4, max 2 sub questions. (Or)	BTL of CO3, At least one sub



10 Appendix 1: Detailed Syllabus Semester –I to Semester VI

Course Title: Digital Electronics Course code: BC23-101 LTP: 3-1-0 Credit : 4

> **Unit 1.** Data and number systems; Binary, Octal and Hexadecimal representation and their conversions; BCD, ASCII, EBDIC, Gray codes and their conversions; Signed binary number representation with 1's and 2's complement methods, Binary arithmetic. Venn diagram, Boolean algebra; Various Logic gates- their truth tables and circuits; Representation in SOP and POS forms; Minimization of logic expressions by algebraic method, K-map method

Unit 2. Combinational circuits- Adder and Subtractor circuits; Applications and circuits of Encoder, Decoder, Comparator, Multiplexer, De-Multiplexer and Parity Generator. [5] Memory Systems: RAM, ROM, EPROM, EEROM, Designof combinational circuits-using ROM, Programming logic devices and gate arrays. (PLAs and PLDs)

Unit 3. Sequential Circuits-Basic memory element-S-R, J-K, D and T Flip Flops, various types of Registers and counters and their design, Irregular counter, State table and state transition diagram, sequential circuits design methodology.

Unit 4. Different types of A/D and D/A conversion techniques. Logicfamilies-TTL,ECL,MOS and CMOS, their operation and specifications.

Course Title: Programming in C Course code: BC23-102 LTP: 3-1-0 Credit: 4

Detailed content

Unit 1: Introduction to Programming

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) - (1 lecture).

Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudo code with examples. (1 lecture)

From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code. (2 lectures)

Unit 2: Arithmetic expressions and precedence

Unit 3: Conditional Branching and Loops

Writing and evaluation of conditionals and consequent branching, Iteration and loops

Unit 4: Arrays

Arrays (1-D, 2-D), Character arrays and Strings.

Unit 5: Basic Algorithms

Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required).

Unit 6: Function

Functions (including using built in libraries), Parameter passing in functions, call by value, passing arrays to functions: idea of call by reference.

Unit 7: Recursion

Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

Unit 8: Structure

Structures, Defining structures and Array of Structures

Unit 9: Pointers

Idea of pointers, Defining pointers, Use of Pointers in self-referential structures, notion of linked list (no implementation).

Unit 10: File handling (only if time is available, otherwise should be done as part of the lab).

Course Outcomes

The student will learn

- To formulate simple algorithms for arithmetic and logical problems. To translate the algorithms to programs (in C language).
- To test and execute the programs and correct syntax and logical errors.
- To implement conditional branching, iteration and recursion.
- To decompose a problem into functions and synthesize a complete program using divide and conquer approach.
- To use arrays, pointers and structures to formulate algorithms and programs.
- To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
- To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration.

Learning Resources:

- 1. R. S. Salaria, Computer Concepts and Programming in C, Khanna Publishers
- 2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
- 3. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill
- 4. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India

Course title: Discrete Mathematics Course code: BC23-103

LTP: 3-1-0

Credit: 4

Boolean algebra: Introduction of Boolean algebra, truth table, basic logic gate, basic postulates of Boolean algebra, principle of duality, canonical form, Karnaugh map.

Abstract algebra: Set, relation, group, ring, field.

Combinatorics: Basic counting, balls and bins problems, generating functions, recurrence relations. Proof techniques, principle of mathematical induction, pigeonhole principle.

Graph Theory: Graphs and digraphs, complement, isomorphism, connectedness and reachability, adjacency matrix, Eulerian paths and circuits in graphs and digraphs, Hamiltonian paths and circuits in graphs and tournaments, trees; Planar graphs, Euler's formula, dual of a planer graph, independence number and clique number, chromatic number, statement of Four-color theorem.

Logic: Propositional calculus - propositions and connectives, syntax; Semantics - truth assignments and truth tables, validity and satisfiability, tautology; Adequate set of connectives; Equivalence and normal forms; Compactness and resolution; Formal reducibility - natural deductions system and axiom system; Soundness and completeness.

Text Books: 1

- Topics in Algebra, I. N. Herstein, John Wiley and Sons.
- Digital Logic & Computer Design, M. Morris Mano, Pearson.
- Elements of Discrete Mathematics, (Second Edition) C. L. LiuMc Graw Hill, New Delhi.
- Graph Theory with Applications, J. A. Bondy and U. S. R. Murty, Macmillan Press, London.
- Mathematical Logic for Computer Science, L. Zhongwan, World Scientific, Singapore.
- Discrete Structures, S.B. Singh, Khanna Publishing House

Reference Books:

- Introduction to linear algebra. Gilbert Strang.
- Introductory Combinatorics, R. A. Brualdi, North-Holland, New York.
- Graph Theory with Applications to Engineering and Computer Science, N. Deo, Prentice Hall, Englewood Cliffs.
- Introduction to Mathematical Logic, (Second Edition), E. Mendelsohn, Van-Nostrand, London.

Course title: PC Software Course code: BC23-104 LTP: 3-1-0 Credit: 4 Introduction MS Windows (Windows '10)

Desktop, creation of folders and shortcuts, features of Windows explorer

Familiarization and using MS packages – Word, Excel, PowerPoint, basic skills in using these tools. (Version MS-Office'2013)

Books:

- 1. Introduction to Computers with MS-Office, Leon, TMH
- 2. Personal Computer Software, EXCEL BOOKS
- 3. A First Course in Computers 2003, Saxena, VIKAS
- 4. Computer Concepts & Windows, Stoline, SPD/LABYRINTH
- 5. Windows'98 in easy steps, Harshad Kotecha, Wiley Dreamtech
- 6. Office 2000 in easy steps, Stephen Copestake, Wiley Dreamtech
- 7. Windows & MS-Office 2000, Krishnan, SCITECH
- 8. Trouble Shooting Microsoft Windows, PHI/MSP

Course code: BC23-192	Category: Engineering Science Courses
Course Title: Programming in C laboratory	Semester: II
L-T-P:0-0-4	Credit: 02

The laboratory should be preceded or followed by a tutorial to explain the approach or algorithm to be implemented for the problem given.

Experiment-I

- To demonstrate the usage of operators and data types in C
- Write a program to print the size of all the data types with its modifiers supported by C and itsrange.
- Write a program to calculate simple interest.

Experiment-II

- To demonstrate the usage of if, if-else, nested-if and switch
- Write a program to find the largest, smallest and second largest of three numbers.
- Write a program to accept marks of three subjects calculate the total percentage and output the result of the student.
- Write a program to find the second largest of four numbers.
- Write a program to calculate Julian date.

Experiment-III

- To demonstrate the usage of while, do-while and for loops
- Write a program to find the sum of numbers from 1 to N.
- Write a program to reverse a number.
- Write a program to generate the Fibonacci series.

Experiment-IV

- To demonstrate the concept of arrays and strings
- Write a program to check whether a string is a Palindrome without using array.
- Write a program to insert a number at a given position in an array.
- Write a program to arrange a list of numbers in ascending order.
- Write a program to check whether a given matrix is symmetric or not.
- Write a program to perform matrix multiplication.

Experiment-V

- To demonstrate the usage of functions and recursion
- Write a program to check whether a given number is prime or not.
- Write a program to find the roots of a quadratic equation
- Write a recursive program to find the factorial of a number.

• Write a recursive program to find sum of natural number using recursion.

Experiment-VI

- To demonstrate the concept of structures
- Write a program to create a student record and display the same.

Experiment-VII

- To demonstrate the concept of pointers
- Write a program using function to swap two numbers using pointers

Experiment-VIII

- To demonstrate the concept of File
- Write a program to create a file and store some records in it. Display the contents of the same. And show the use of write, read and append mode.

*Note: The above are only suggestive programs. Any other programs can be added as per requirements by the faculty.

Course title: PC Software Laboratory

Course code: BC23-194

LTP: 3-1-0

Credit: 4

Basic skills lab

Introduction to MS Office, Windows Overview, Office features, Templates and Wizards, MS Word, Excel, PowerPoint, Outlook (Overview), MS Access (Table, relation, queries, reports)

(Ms – Office Version 2013)

Course title: Computer Architecture & Organization Course code: BC23-201 LTP: 3-1-0 Credit: 4 Revision of basics in Boolean logic and Combinational/Sequential Circuits.

Functional blocks of a computer: CPU, memory, input-output subsystems, control unit.

Instructionsetarchitecture of a CPU: Registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Outlining instruction sets of some common CPUs.

Data representation: Signed number representation, fixed and floating point representations, character representation.

Computer arithmetic: Integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and-add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic, IEEE 754 format.

Introduction to x86 architecture.

CPU control unit design: Hardwired and micro-programmed design approaches, design of a simple hypothetical CPU.

Memory system design: Semiconductor memory technologies, memory organization.

Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O Transfers– program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes

- Role of interrupts in process state transitions, I/O device interfaces - SCII, USB.

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards. Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Text Books:

1. Computer System Architecture M. M. Mano:, 3rd ed., Prentice Hall of India, New Delhi, 1993.

2. Computer Organization and Design: The Hardware/Software Interface, David A. Patterson and John L. Hennessy.

3. Computer Organization and Embedded Systems, Carl Hamacher.

Reference Books:

- 1. Computer Architecture and Organization, John P. Hayes.
- 2. Computer Organization and Architecture: Designing for Performance, William Stallings.
- 3. Computer System Design and Architecture, Vincent P. Heuring and Harry F. Jordan

Course code: BC23-202	Category:	
Course Title: Data Structure using C	Semester: III	
L-T-P : 3-1-0	Credit: 4	

Detailed Contents

Basic Terminologies and Introduction to Algorithm & Data Organisation: Algorithm specification, Recursion, Performance analysis, Asymptotic Notation - The Big-O, Omega and Thetanotation, Programming Style, Refinement of Coding-Time-Space Trade Off, Testing, Data Abstraction

Linear Data Structure: Array, Stack, Queue, Linked-list and its types, Various Representations, Operations& Applications of Linear Data Structures

Non-linear Data Structure: Trees (Binary Tree, Threaded Binary Tree, Binary Search Tree, B & B+ Tree, AVL Tree, Splay Tree) and Graphs (Directed, Undirected), Various Representations, Operations & Applications of Non-Linear Data Structures

Searching and Sorting on Various Data Structures: Sequential Search, Binary Search, Comparison Trees, Breadth First Search, Depth First Search Insertion Sort, Selection Sort, Shell Sort, Divide and Conquer Sort, Merge Sort, Quick Sort, Heap sort, Introduction to Hashing

File Organization (Sequential, Direct, Indexed Sequential, and Hashed) and various types of accessing schemes.

Graph: Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

Text Books:

- 1. Fundamentals of Data Structures, E. Horowitz, S. Sahni, S. A-Freed, UniversitiesPress.
- 2. Data Structures, R.S. Salaria, Khanna Book Publishing, Delhi.
- 3. Data Structures and Algorithms, A. V. Aho, J. E. Hopperoft, J. D. Ullman, Pearson.
- 4. Expert Data Structures with C, R.P. Patel, Khanna Publishing House.

Reference Books:

1. The Art of Computer Programming: Volume 1: Fundamental Algorithms, Donald E. Knuth.

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

3. Introduction to Algorithms, Thomas, H. Cormen, Charles E. Leiserson,

Course Title : Mathematics-I Course Code : BC23-203 LTP: 3-1-0 Credit: 4

Module **Description of Topic** No. Calculus (Integration): Evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate 1 surface areas and volumes of revolutions. Calculus (Differentiation): Rolle's Theorem, Mean value theorems, Taylor's and Maclaurin's theorems with remainders; Indeterminate forms and L'Hospital's rule; Maxima and minima. 2 Matrices: Matrices, Vectors: addition and scalar multiplication, matrix multiplication; Linear systems of equations, linear Independence, rank of a matrix, determinants, 3 Cramer's Rule, inverse of a matrix, Gauss elimination and Gauss-Jordan elimination. Vector Spaces: Vector Space, linear dependence of vectors, Basis, Dimension; Linear transformations (maps), Range and Kernel of a linear map, Rank and Nullity, Inverse 4 of a linear transformation, Rank-Nullity theorem, composition of linear maps, Matrix associated with a linear map. Vector Spaces (Continued): Eigenvalues, Eigenvectors, Symmetric, Skew-symmetric, and Orthogonal Matrices, Eigenbases. 5 Diagonalization; Inner product spaces, Gram-Schmidt orthogonalization.

Course Outcomes:

The students will be able to:

- Apply the concept and techniques of differential and integral calculus to determine curvature and evaluation of different types of improper integrals.
- Understand the domain of applications of mean value theorems to engineering problems.
- Learn different types of matrices, concept of rank, methods of matrix inversion and their applications. Understand linear spaces, its basis and dimension with corresponding applications in the field of computer science

- - Learn and apply the concept of eigen values, eigen vectors, diagonalisation of matrices and orthogonalization in inner product spaces for understanding physical and engineering problems

Learning Resources:

- 1. Reena Garg, Engineering Mathematics-I, Khanna Publishers.
- 2. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.
- 3. Michael Greenberg, Advanced Engineering Mathematics, Pearson.
- 4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
- 5. Kanti B. Dutta, Mathematical Methods of Science and Engineering, Cenage Learning.
- 6. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi.
- 7. S.K. Mapa, Higher Algebra: Abstract and Linear, Sarat Book House Pvt.Ltd.
- 8. Hoffman and Kunze: Linear algebra, PHI.

Course Code : BC23-204	Category :	
Course Title : English	Semester : Second	
L-T-P : 3-1-0	Credit:4	
Pre-Requisites:		

Detailed contents

1. Vocabulary Building

- 1.1 The concept of Word Formation: Compounding, Backformation, Clipping, Blending.
- 1.2 Root words from foreign languages and their use in English
- 1.3 Acquaintance with prefixes and suffixes from foreign languages in English to form derivatives.
- 1.4 Synonyms, antonyms, and standard abbreviations: Acronyms

2. Basic Writing Skills

- 2.1 Sentence Structures & Types: Simple, Compound, Complex
- 2.2 Use of phrases and clauses in sentences: Transformation of sentences, active, passive, narration
- 2.3 Importance of proper punctuation
- 2.4 Creating coherence: Arranging paragraphs & Sentencesin logical order
- 2.5 Creating Cohesion: Organizing principles of paragraphs indocuments
- 2.6 Techniques for writing precisely

3. Identifying Common Errors in Writing

- 3.1 Subject-verb agreement
- 3.2 Noun-pronoun agreement
- 3.3 Misplaced modifiers
- 3.4 Articles
- 3.5 Prepositions
- 3.6 Redundancies
- 3.7 Clichés

4. Nature and Style of sensible Writing

- 4.1 Describing
- 4.2 Defining
- 4.3 Classifying
- 4.4 Providing examples or evidence
- 4.5 Writing introduction and conclusion

5. Writing Practices

- 5.1 Comprehension
- 5.2 Précis Writing
- 5.3 Essay Writing
- 5.4 Business Letter, Cover Letter & CV; E-mail

Learning Resources:

- (i) Kulbushan Kumar, R S Salaria, Effective Communication Skills, Khanna Publishing House, Delhi.
- (ii) Practical English Usage. Michael Swan. OUP. 1995.
- (iii) Remedial English Grammar. F.T. Wood. Macmillan.2007
- (iv) On Writing Well. William Zinsser. Harper Resource Book. 2001
- (v) Study Writing. Liz Hamp-Lyons and Ben Heasly. Cambridge University Press. 2006.
- (vi) Communication Skills. Sanjay Kumar and PushpLata. Oxford University Press. 2011.
- (vii) Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press
- (viii) Universal English Prof. Prasad Kataria Publications, 2019.
- (ix) "Communication Skills for Professionals"-Nira Konar, Prentice Hall of India 2nd edition, New Delhi, 2011
- (x) Gajendra Singh Chauhan, Smita Kashiramka and L. Thimmesha. Functional English. Cengage , 2019.

Course Outcomes

The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills.

Course code: BC23-292	Category:
Course Title: Data Structure using C Laboratory	Semester: II
L-T-P : 0-0-4	Credit: 2

The laboratory should be preceded or followed by a tutorial to explain the approach or algorithm to be implemented for the problem given.

List of Practical:

- 1. Implementation of searching and sorting techniques.
- 2. Implementation of linkedlist.
- 3. Implementation of push and pop operation on stack
- 4. Implementation of inqueue and dequeue operation.
- 5. Write a program to solve the problems using iteration/recursion
- 6. WriteaprogramtoimplementMergeSort
- 7. Write a program to implement Bubble Sort
- 8. Program for storing data as tree structure and implementation of various traversal techniques
- 9. Program for storing data as graph structure and implementation of various traversal techniques
- 10. Write a program to implement Linear Sort
- 11. Write a program to implement Binary Sort

** Faculty may add or remove experiments as per syllabus requirements.

Course Title : English Language and Language Laboratory-I Course code: BC23-294 LTP: 0-0-4 Credit: 2

- 1) Honing 'Listening Skill' and its sub skills through Language Lab Audio device;
- 2) Honing 'Speaking Skill' and its sub skills
- 3) Helping them master Linguistic/Paralinguistic features(Pronunciation/Phonetics/ Voice modulation/ Stress/ Intonation/ Pitch &Accent) of connected speech
- 4) Honing 'Conversation Skill' using Language Lab Audio --Visual

input; Conversational Practice Sessions (Face to Face / via

Telephone, Mobile phone & Role Play Mode)

5) Introducing 'Group Discussion' through audio –Visual input and acquainting them with key strategies for success

Course Out Course Title Course cod LTP: 3-1-0 Credit: 4 Unit	 7) Honing 'Reading Skills' and its sub skills using Visual / Graphics/ Diagrams /Chart Display/Technical/Non Technical Passages/Learning Global / Contextual / Inferential Comprehension; 8) Honing 'Writing Skill' and its sub skills by using Language Lab Audio –Visual input; Practice Sessions tcomes The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills. e: Software Engineering le: BC23-301
Course Out Course Title Course cod LTP: 3-1-0 Credit: 4 Unit	 8) Honing 'Writing Skill' and its sub skills by using Language Lab Audio –Visual input; Practice Sessions tcomes The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills. e : Software Engineering le: BC23-301 Conten t
Course Out Course Title Course cod LTP: 3-1-0 Credit: 4 Unit	 The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills. e : Software Engineering le: BC23-301 Conten t
Course Title Course cod LTP: 3-1-0 Credit: 4 Unit	 The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills. e : Software Engineering le: BC23-301 Conten t
Course Title Course cod LTP: 3-1-0 Credit: 4 Unit	comprehension, writing and speaking skills. e : Software Engineering le: BC23-301 Conten t
Course Title Course cod LTP: 3-1-0 Credit: 4 Unit	e : Software Engineering le: BC23-301 Conten t
Course cod LTP: 3-1-0 Credit: 4 Unit	le: BC23-301 Conten t
LTP: 3-1-0 Credit: 4 Unit	Conten t
Unit	Conten t
	Introduction: Programming in the small vs. programming in the large: software project failur
1 and importance of software quality and timely availability: of software engineering toward	
	successful execution of large software projects; emergence of software engineering as a
	discipline, Software Engineering Historical Development from Jackson Structured
	Programming to Agile Development.
	Software Project Management: Basic concepts of life cycle models – different models and
2	milestones; software project planning –identification of activities and resources; concepts of
	teasibilitystudy; techniques for estimation of schedule and effort; software cost estimation models and concepts of software orginacting occupations; techniques of software project
	controland reporting: introduction to measurement of software size: introduction to the
	concepts of risk and its mitigation; configuration management.
3	Software Quality Management and Reliability: Software quality; Garvin's quality dimension
	McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction
	Capability Maturity Models (CMM and CMMI); Introduction to software reliability
	Software Requirements Analysis Design and Construction: Introduction to
4	Software Requirement Specifications (SRS) and requirement elicitatic
	techniques; techniques for requirement modelling– decision tables, event tables, state
	transition tables Petrinets; requirements documentation through use cases; introduction t
	UML, introduction to software metrics and metrics-based control methods; measure of code
3	Software Quality Management and Reliability: Software quality; Garvin's quality dimenses McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction Capability Maturity Models (CMM and CMMI); Introduction to software reliability reliability models and estimation. Software Requirements Analysis, Design and Construction: Introduction to Software Requirement Specifications (SPS) and requirement elicity

5 Object Oriented Analysis, Design and Construction: Concepts -- the principles of abstraction, modularity, specification, encapsulation and information hiding; concepts of abstract data type; Class Responsibility Collaborator (CRC) model; quality of design; design measurements; concepts of design patterns; Refactoring; object-oriented construction principles; object oriented metrics. 6 Software Testing: Introduction to faults and failures; basic testing concepts; concepts of verification and validation; black box and white box tests; white box test coverage – code coverage, condition coverage, branch coverage; basic concepts of black-box tests equivalence classes, boundary value tests, usage of state tables; testing use cases; transaction based testing; testing for non-functional requirements – volume performance and efficiency; concepts of inspection; Unit Testing, Integration Testing, System Testing and Acceptance Testing. 7 Agile Software Engineering: Concepts of Agile Methods, Extreme Programming; Agile ProcessModel

Scrum, Feature; Scenarios and Stories

Course code: BC23-302	Category:
Course Title: Operating Systems	Semester: III
L-T-P : 3-1-0	Credit: 4

Detailed Content

Unit I.

OverviewofOperatingSystems:OSandtheComputerSystem,Efficiency,SystemPerformance and User Convenience, Classes of Operating Systems, Batch Processing Systems, Multiprogramming Systems, Time Sharing Systems, Real Time Operating Systems, Distributed Operating Systems, Modern Operating Systems.

Unit II.

Processes and Threads: Processes and Programs, Programmer view of Processes, OS view of Processes, Threads, Case studies of Processes and Threads.

Scheduling: Preliminaries, Non-preemptive Scheduling Policies, Preemptive Scheduling Policies, Scheduling in Practice, Real Time Scheduling, Scheduling in Unix, Scheduling in Linux, Scheduling in Windows, Performance Analysis of Scheduling Policies.

Unit III.

Memory Management: Managing the Memory Hierarchy, Static and Dynamic Memory Allocation, Memory Allocation to a Process, Reuse of Memory, Contiguous Memory Allocation, Noncontiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Kernel Memory Allocation, Virtual Memory: Virtual Memory Basics, Demand Paging, Page Replacement Policies, Memory Allocation to a Process, Shared Pages, Memory Mapped Files, Unix Virtual Memory, Linux Virtual Memory, Virtual Memory using Segmentation.

Unit IV.

Security and Protection: Overview of Security and Protection, Goals of Security and Protection, Security Attacks, Formal and Practical aspects of Security, Encryption, Authentication and Password Security, Access Descriptors and the Access Control Matrix, Protection Structures, Capabilities, Unix Security, Linux Security, WindowsSecurity

TEXT BOOK:

- Silberschatz et.al.-Operating System Concepts, 6th Edition, John Wiley Inc., 2003
- H.M. Deitel -Operating Systems, 6th Edition, Pearson Education, 2006
- Robbins-Linux Programming by Example-Pearson Education, New Delhi-2005
- Sumitabh Das : Your UNIX The Ultimate Guide; TMH

REFERENCE BOOKS:

- D.M. Dhandhare Operating Systems, 2nd Edition, Tata McGraw Hill, New Delhi, 2006
- J.Goerzen-Linux Programming Bible, IDG Books, New Delhi-2001
- N.Mathew & R.Stones-Beginning Linux Programming Wiley Publishing India, 2004.
- S.E. Mandnick & J.J. Donovan : Operating System; TMH

Course Title : Computer Graphics and Internet Course code: BC23-303 LTP: 3-1-0 Credit: 4

Computer graphics - Co-ordinate systems, Homogenous co-ordinate systems, line drawing algorithms, circle drawing algorithms, Two- dimension transformations (rotation, scaling, sharing etc).

Raster scanning, CRT (Interface Design)

Clipping Algorithm (Sudderland-cohen line clipping Algorithm), Projection (Two-dimensional), Bazier, B-spline curves, shadowing. TCP/IP, addressing in Internet – IP and domains, Servers, type of Connectivity.

Email services and protocols (X400, SMTP, UUCD, PPP, POP), FTP.

Web publishing - HTTP, browsers (naming), Introduction to HTML, Java script, use of Java applets within HTML files, ASP (Cookies and database connectivity only).

Internet security, Introduction to e-commerce, electronic payment standards and methods.

Books:

1. Procedural & Mathematical Elements in Computer Graphics, Rogers, TMH

- 2. Computer Graphics, Hearn & Baker, PHI
- 3. Computer Graphics, EXCEL BOOKS
- 4. Introduction to Computer Graphics, A. Mukherjee, VIKAS 5.Fundamentals of Computer Graphics & Multimedia, Mukherjee, PHI6.Beginning ASP 3.0, Baser, SPD/WROX
- 5. Dynamic HTML, Goodman, SPD/O'REILLY
- 6. HTML Black Book , Stephen Holzner, Wiley Dreamtech 9.ASP 3.0 programming Bible,Eric A. Smith, Wiley Dreamtech 10.Computer Graphics, Bhandari & Joshi, EPH

Course Title : OOPS using C++ Course code: BC23-304 LTP: 3-1-0 Credit: 4 <u>Detailed content</u>

Procedural programming, An Overview of C: Types Operator and Expressions, Scope and Lifetime, Constants, Pointers, Arrays, and References, Control Flow, Functions and Program Structure, Namespaces, error handling, Input and Output (C-way), Library Functions (string, math, stdlib), Command line arguments, Pre-processor directive

SomedifferencebetweenCandC++:Singlelinecomments, Local variable declaration within function scope, function declaration, function overloading, stronger type checking, Reference variable, parameter passing–valuevs. reference, passing pointer by value or reference, #define constant vs const, Operator new and delete, the type casting operator, Inline Functions in contrast tomacro, default arguments

The Fundamentals of Object Oriented Programming: Necessity for OOP, Data Hiding, Data Abstraction, Encapsulation, Procedural Abstraction, Class and Object.

More extensions to C in C++ to provide OOP Facilities: Scope of Class and Scope Resolution Operator, Member Function of a Class, private, protected and public Access Specifier, this Keyword, Constructors and Destructors, friend class, error handling (exception)

Essentials of Object Oriented Programming: Operator overloading, Inheritance – Singleand Multiple, Class Hierarchy, PointerstoObjects, AssignmentofanObjecttoanotherObject, Polymorphismthroughdynamic binding, Virtual Functions, Overloading, overriding and hiding, Error Handling.

Generic Programming: Template concept, class template, function template, template specialization.

Input and Output: Streams, Files, Library functions, formatted output.

Text Books:

• The C++ Programming Language, Bjarne Stroustrup, Addison Wesley.

• C++ and Object-Oriented Programming Paradigm, Debasish Jana, PHI Learning Pvt. Ltd. Reference Books:

- Programming Principles and Practice Using C++, Bjarne Stroustrup, Addison Wesley.
- The Design and Evolution of C++, Bjarne Stroustrup, Addison Wesley.

Course Title : Management and Accounting Course code: BC23-305 LTP: 3-1-0 Credit: 4

Basics of management; Planning, scheduling, organising, staffing, directing, controlling Managerial economics and financial management, productivity management

Financial accounting, financial statements and analysis Conceptual framework of cost accounting

Cost-volume profit relationship, budgeting, cost accumulation system, variable and absorption costing system Financial accounting computer packages

Financial Management-Finance functions in Business. Relation of finance with other functions.

Source of finance long term and short term. Financial institution – IDBI, ICICI, IFCI and Commercial Banks.

Conceptual framework of Cost-Accounting- Basic cost concept. Cost determination process, costing for materials, labour and overheads. Profitability Analysis – budgeting – application of Capital budgeting techniques for decision making.

Books:

1.Management Accounting, M.E.Thukaram Rao, New Age International 2.Management Accounting, Khan & Jain, TMH

3.Cost Accounting-An Introduction, Nigam & Jain, PHI 4.Management Accounting, Pande, VIKAS

Accounting and Financial Management for MCA & MBA students, SCITECH

Management Accounting, A.P.Rao. EPH.

Cost & Management Accounting ,Inamdar.EPH

Course Title : Operating system & Shell programming Course code: BC23-392 LTP: 0-0-4

Credit: 2

The laboratory should be preceded or followed by a tutorial to explain the approach or algorithm to be implemented for the problem given.

- 1. Implement in C the following UNIX commands using System calls: cat and mv.
- 2. Write a program in C to determine the size of a file using the lseek command.
- 3. Write a program to calculate the number of blocks assigned for the file.
- 4. Write a C program that deletes a directory with all its subfolders. The name of the directory should be read from the command line.
- 5. Write a program that deletes every 5th byte from a file, but without using a temporary file or allocating a buffer in the memory.
- 6. Write a program in C to implement FCFS CPU scheduling Algorithm.
- 7. Write a program in C to implement SJF CPU scheduling Algorithm.
- 8. Write a program in C to implement Priority CPU scheduling Algorithm.
- 9. Write a program in Cto implement Round Robin (RR) CPU scheduling Algorithm.
- 10.Write a program in c to read from the buffer & produce desired output.
- 11. Write a program in C to create Userid & Password.
- 12. Write a program in c to implement and find how many Users currently login in NetWork.
- 13. Write a program in c to create your own system call just like a copy. Write a program in c to create your won system call just like a delete.
- 14. Write a program in c to find the Disk Space.
- 15. Write a program In C to find The number of pages in the process.
- 16. Write a program In C to find The number of frames allocated to the process.
- 17. Write a program in c to find the no. of blocks occupied by a file.
- 18.Write a program in c to create your won system call just like a delete.
- 19. Write a program in c to create your won system call just like a ls.
- 20. Write a program in c to find a PID no. of any Process ** Faculty may add or remove experiments as per syllabus requirements.

Course Title : OOPS using C++ Laboratory Course code: BC23-394 LTP: 0-0-4 Credit: 2 Exercise 1

- Write a C++ program to display "Hello, Welcome to C++ Programming"
- Write a C++ program to print details name, roll number in a single and two lines.
- Write a C++ program to print name by reading, assigning and initializing to a variable with an appropriate prompt.
- Write a C++ program to print your personal details name, surname (single character),total marks, gender(M/F), result(P/F) by taking input from the user.

Exercise 2

- Write a C++ program to convert centigrade into Fahrenheit. Formula: C=(F-32)/1.8
- Write a C++ program that declares two integers, determines whether the first is a multiple of the second and print the result. (Hint: Use the remainder operator)
- Write a C++ program that prompts the user to enter two integer values in int variables val1, val2 and find largest, sum, difference, product and ratio of these values.
- WriteaC++program that prompts the user to enter three integer values, and then outputs the values in numerical sequence separated by commas. So, if the user enters the values 1046, the output should be 4,6,10. If two values are the same, they should just be ordered together. So, the input 454 should give 4,4,5.

Exercise 3

- Write a C++ program to read a sequence of double values into a vector. Think of each value as the distance between two cities along a given route. Compute and print the total distance. Find and print the smallest and greatest distance between two neighboring cities. Find and print the mean distance of the neighboring cities.
- Write a C++ program to convert decimal to binary.
- Write a C++ program to print the accepted number and its reverse number.
- Write a C++ program to read marks in 6 subjects using vectors and find average marks **Exercise 4**
 - Write a C++ program to read names using vector and display the names and their count.
 - Write a C++ program to read a number between 1 and 100 and the program asks questions to figure out what the number is (e.g. "Is the number you are thinking of less than 50?"). Your program should be able to identify the number after asking no more than seven questions. Hint: use the < and <= operators and the if-else statement.
- Develop a simple calculator using if-else if and switch-case.

Exercise 5

- Write a C++ program to find the sum of individual digits of a positive integer.
- Make a vector holding the ten strings "zero", "one", ..., "nine". Use that in a program that converts a digits to its corresponding spelled out values; e.g. the input 5 gives the output five. Have thesame program, using the same input loop, convert spelled out numbers into their digit form; e.g., the input five gives the output 5.
- Write a C++ program to find all the prime numbers between 1 and 100. Write a function to do this.

Exercise 6

- Write a C++ program that uses functions.
- To swap twointegers

- To swap characters
- To swap tworeals
- Write a C++ program that reads a series of numbers and stores them in a vector <int>. After the user inputs all the numbers he wishes to, ask how many of the numbers the user wants to sum. For an answer N, print the sum of the first Nelements of the vector. For example "Please entersome numbers (press '0' at prompt to stop):"
- 1223132415 "Please enter how many of the numbers you wish to sum, starting from the first:" 3 "The sum of the first 3 numbers: 1223 and 13 is 48"
- Write a C++ program that writes out the Fibonacci series. Find the largest Fibonacci number that fits in an int.

Exercise 7

- Define a class name_value that holds a string and a value. Give it a constructor (a bit like Token). Use vector<name_value> instead of two vectors.
- Create a class employee that includes firstname(type String), lastname(type String) and a monthly salary. Create two employee objects and display each object's yearly salary. Give each employee a 10; raise and display each employee's yearly salary.

Exercise 8

- Write a C++ program that reads digits and computes them into integers. For example 123 is read as the characters 1,2 and 3. The program should output "123 is 1 hundred and 2 tens and 3 ones". The number should be output as an int value. Handle numbers with one, two, three or four digits. Hint: to get the integer value 5 of the character '5' subtract '0' that is '5'-'0'==5.
- Provide name constants that you really can't change the value of. Hint: you have to add a member to variable that distinguishes between constants and variables and check for it in set_value(). If you want to let the user define constants. You'll have to add a notation to let the user express that, for example, const pi=3.14.

Exercise 9

- Write a function print() that prints a vector of ints to cout. Give it two arguments; a string for "labeling" the output and a vector.
- Write two functions that reverse the order of elements in a vector<int>. The first reverse function should produce a new vector with the reversed sequence, leaving its original vector unchanged. The other reverse function should reverse the elements of its vector without using any other vectors.

Exercise 10

- Write a function randint() that produces a pseudo-random number in the range [0:MAXINT].
- Write a function that using randint() from the previous exercise. Computes a pseudorandom integer in the range(a:b). rand_in_range(int a, int b).
- Write a function that finds the smallest and the largest element of a vector argument and also computes the mean and the median. Do not use global variables. Either return a struct containing the results or pass them back through reference arguments.

Exercise 11

• Write a function that takes a vector<string> argument and returns a vector<int> containing the number of characters in each string. Also find the longest and the shortest string and the lexicographically first and last string.

• Write a function that given two vector<double> price and weight computes a value (an "index") that is the sum of all price[i]*weight[i]. Note that we must have weight.size()<=price.size().

Exercise 12

- Write a C++ program to display the contents of a text file.
- Write a C++ program that counts the characters, lines and words in the text file.
- Write a C++ program that produces the sum of all the numbers in a file of whitespace separated integers.

Exercise 13

- Write a C++ program that creates a file of data in the form of the temperature. Fill the file with at least 50 temperature readings. Call this program store_temps.cpp and the file it creates raw_temps.txt.
- Write a C++ program that accepts two file names and produces a new file that is the contents of the first file followed by the contents of the second; that is, the program concatenates the two files.

Exercise 14

- Write a C++ program that given a file name and a word outputs each line that contains that word together with the line number. Hint:getline().
- $\bullet \quad Write a C ++ program that reads a text file and converts its input to all lower case, producing a new file.$

Exercise 15

- Write a C++ program that replaces punctuation with whitespace. For example, "don't use the as-if rule" becomes dont use the asif rule".
- Write a C++ program to reverse the order of characters in a text file. For example, asdfghjkl becomes lkjhgfdsa.

Exercise 16

- Write a C++ program that reads a text file and writes out how many characters of each character classification are in the file.
- Write a C++ program draw a rectangle as a rectangle and as a polygon. Make the lines of the polygon red and the lines of the rectangle blue.

Exercise 17

- Write a C++ program to draw the Olympic five rings.
- Write a C++ program to display an image like photo on the screen.
- Write a C++ program to draw a part of an ellipse by defining a class arc. Hint: fl_arc().

Exercise 18

- Write a C++ program to draw a box with rounded corners. Define a class box, consisting of four lines and four arcs.
- Write a C++ program to draw a line with an arrowhead by defining a class arrow.

Exercise 19

• Define a class poly that represents a polygon but checks that its points really do make a polygon in its constructor. Hint: you'll have to supply the points to the constructor.

- Defineaclassstar. Oneparameter should be the number of points. WriteaC++program to drawa few stars with differing numbers of points, differing line colors, and differing fill colors.
- Define two classes smiley and frowny, which are both derived from class circle and have two eyes and amouth. Next, derive classes from smiley and frowny, which add an appropriate hattoeach.

Exercise 20

- Write a C++ program to write a function void to_lower(char* s) that replaces all uppercase characters. Don't use any standard library functions.
- Write a C++ program to write a function, char* findx(const char* s, const char* x), that finds the first occurrence of the string x in s.

Exercise 21

- Write a C++ program that reads characters from cin into an array that you allocate on the free store. Read individual characters until an asterisk (*) is entered. Do not use a std::string.
- Write a C++program to write a function, char* strdup(const char*) that copies a string into memory it allocates on the free store. Use the dereference operator * instead.

Exercise 22

• Write a C++ program to write a function char* findx(const char* s, const char* x) that find the first occurrence of the string x in s. Use dereference operator * instead.

Exercise 23

- Write a C++ program to write a function template for finding the minimum value contained in any array.
- Writeatemplatefunctionthattakesavector<T>vtandavector<U>vuasargumentsand returnsthe sum of allvt[i]*vt[i]s.
- Define a class Int having a single member of class int. Define constructors, assignment, and operators +,-,*,/ for it.
- Implement vector::operator=() using an allocator for memory management.

Exercise 24

- Define a file handle class with a constructor that takes a string argument (filename), opens the file in the constructor, and closes it in the destructor.
- Define an input and an output operator (>> and <<) for vector.
- Given a list<int> as a (by-reference) parameter, make a vector<double> and copy the elements of the list into it. Verify that the copy was complete and correct. Then print the elements sorted in order of increasing value.
- ** Faculty may add or remove experiments as per syllabus requirements.

Course Title: Internet & Computer Graphics Lab Course code: BC23-393 LTP:0-0-4 Credit: 2 Developing web pages with HTML, Using ASP, Creating and experimenting with computer graphics. (with C- Language)

** Faculty should add experiments as per syllabus requirements and maintain a laboratory manual.

Course Title: Data Base Management Systems Course code: BC23-401 LTP:3-1-0 Credit: 4

Detailed content

UNIT-I: Database system architecture:

Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationshipmodel, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

Unit-II: Relational query languages and Relational algebra and calculus:

Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQLserver.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimizational gorithms.

Unit-III: Storage strategies: Indices, B-trees, hashing.

Unit-IV TRANSACTION PROCESSING:

Concurrencycontrol, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

Unit-V Database Security:

Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

Unit-VI Advanced topics:

Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Course Title: Numerical Methods Course code:- BC23-402 LTP: 3-1-0 Credit: 4

Unit: 1 Approximation in numerical computation: Truncation and rounding errors, Fixed and floating- point arithmetic, Propagation of errors.

Unit2: Interpolation: Newtonforward/backward interpolation, Lagrange's and Newton's divided difference Interpolation.

Unit 3: Numerical integration: Trapezoidal rule, Simpson's1/3 rule, Expression for corresponding error terms.

Unit4:Numerical solution of a system of linear equations: Gausselimination method, Matrix inversion, LU Factorization method, Gauss-Seideliterative method

Unit 5: Numerical solution of Algebraic equation:

Bisection method, Regula-Falsi method, Newton-Raphson method

Unit 6: Numerical solution of ordinary differential equation: Euler's method, Runge-Kutta methods, Predictor-Corrector methods and Finite Difference method.

Text book and Reference books:

- R.S. Salaria: Computer Oriented Numerical Methods, Khanna Publishing House
- C.Xavier: C Language and Numerical Methods.
- Dutta & Jana: Introductory Numerical Analysis.
- J.B.Scarborough: Numerical Mathematical Analysis.
- Jain, Iyengar, & Jain: Numerical Methods (Problems and Solution).
- Balagurusamy: Numerical Methods, Scitech.
- Baburam: Numerical Methods, Pearson Education.
- N. Dutta: Computer Programming & Numerical Analysis, Universities Press.

Course Title: Web Technology Course code: BC23-403 LTP: 3-1-0 Credit: 4

Web Development:

HTML, Structure, Tags, Lists, Table, Link and it's types ,Images, Form, Frame, Style sheets and it's type

Introduction to Java:

Java and Java applications, Java Virtual Machine(JVM), Java Runtime Environment(JRE)Java Development Kit(JDK,) Byte code, Java characteristics, Object oriented Programming, Simple java programs, Data types, Operators, Expressions, control statements, Selection statements, Iteration statements, Jump statements

Classes, Inheritance :

Classes in java, Declaring a class, Creating instances of class, Constructors, Argument Passing, use of static keyword, Innner class. Method overloading, Inheritance, use of super keyword ,Method overriding, Abstract class, Dynamic method dispatch, use of final keyword

Interface, Package:

Package, Acesss control mechanism, Interface, Dynamic Method look up

Exception Handling:

Java Exception Handling Mechanism, try, catch, throw, throws and finally, Exception types, Built in Exceptions: checked and unchecked exceptions, User defined Exceptions

String Handling:

String and String Buffer, Constructors, String operations : character extractions, String comparisons, searching, strings, modifying a string. To String() and valueOf() methods, String Buffer operations

Java I/O Stream:

I/O basics, Byte stream, Character stream, Reading console input, Writing console output, Reading and writing files

Java Utility package:

Collection overview, Collection interfaces, Collection classes: ArrayList, LinkedList, Accessing a collection using, iterator and for-Each statement

Applet:

Applet class, Applet architecture, Applet Skeleton, Life cycle methods, setForeground() and set Bachground()methods, Using the status window,HTML Applet tag, Passing parameters to an applet, GetCodebase() and Get Documentbase() methods.

Event Handling and AWT:

Delegation Event Model, Event classes, Sources of Events, Event Listener interfaces, Event handling using adapter class, Inner and anonymous class, AWT classes: Label,Button,TextField etc.

JDBC: Connectivity to database

Course Title: Values and Ethics in Profession Course code: BC23-404 LTP: 3-1-0 Credit: 4 Science, Technology and Engineering as Knowledge and as Social and Professional Activities (2 lectures) Effects of Technological Growth: Rapid Technological growth and depletion of resources. Reports of the Club of Rome. Limits of growth; sustainable development(2 lectures) Energy Crisis; Renewable Energy Resources (2 lectures) Environmental degradation and pollution. Eco-friendly Technologies. Environmental Regulations. Environmental Ethics(4 lectures) Appropriate Technology Movement of Schumacher: later developments (2 lectures) Technology and developing nations. Problems of Technology transfer. Technology assessment, impact analysis (4 lectures) Human Operator in Engineering projects and industries. Problems of man machine interaction. Impact of assembly line and automation. Human centered Technology (4 lectures) **Ethics of Profession** Engineering profession: Ethical issues in engineering practice. Conflicts between business demands and professional ideals. Social and ethical responsibilities of Technologists. Codes of professional ethics. Whistle blowing and beyond. Case studies. (8 lectures) Profession and Human Values Value Crisis in contemporary society (2 lectures) Nature of values: Value Spectrum of a 'good' life (2 lectures) Psychological values: Integrated personality; mental health (2 lectures) Societal values: The modern search for a 'good' society, justice, democracy, secularism, rule of law; values in Indian Constitution. (4 lectures) Aesthetic values: Perception and enjoyment of beauty, simplicity, clarity (2 lectures) Moral and ethical values: Nature of moral judgments; canons of ethics; ethics of virtue; ethics of duty; ethics of responsibility **Books:** Ethics in Mgmt & Indian Ethos, Ghosh, VIKAS

- Business Ethics, G. Pherwani, EPH.
- Ethics, Indian Ethos & Mgmt, Balachandran, Raja & Nair, SHROFF Publishers

Course Title: Database Management System Laboratory Course code: BC23-491 LTP:0-0-4 Credit: 2

Structured Query Language

- 1. Creating Database
- Creating a Database
- Creating a Table
- Specifying Relational Data Types
- Specifying Constraints
- Creating Indexes

2. Table and Record Handling

- INSERT statement
- UsingSELECT and INSERT together
- DELETE, UPDATE, TRUNCATEstatements
- DROP, ALTER statements

3. Retrieving Data from a Database

- The SELECT statement
- Using the WHEREclause
- Using Logical Operators in the WHERE clause
- Using IN, BETWEEN, LIKE, ORDER BY, GROUP BY and HAVING Clause
- Using Aggregate Functions
- Combining Tables Using JOINS
- Subqueries
- 4. Database Management
- Creating Views
- Creating Column Aliases
- 5. Creating Database Users and using Grant and Revoke

Course Title: Numerical Methods using C Laboratory Course code: BC23-492

LTP:0-0-4

Credit-2

- 1. Solving various problems related to object oriented programme with C++
- 2. Implement Numerical problems Using C/MAT LAB
- 3. Assignments on Interpolation: Newton forward &backward, Lagrange
- 4. Assignments on Numerical Integration: Trapezoidal Rule, Simson's 1/3 Rule, Weddle's Rule
- 5. Assignments on Numerical solution of a system of linear equation: Gauss elimination, Gasuss Jacobi, Matrix Inversion, Gauss Seidel
- 6. Assignments on Algebraic Equation: Bisection, Secant, Regula-falsi, Newton Raphson
- 7. Assignments on Ordinary Differential Equation: Taylor Series, Euler's method,
- ** Faculty may add experiments as per syllabus requirements.

Course Title: Web Technology Course code: BC23-493 LTP:0-0-4 Credit: 2

Programming Lab of JAVA

- 1. Write a program to input two numbers and find its sum.
- 2. Write a program to input principal amount, rate and time. Calculate its simple interest and compound interest.
- 3. Write a program to input the radius of any circle and find its area and perimeter.
- 4. Write a program to input a number and check it is positive or negative.
- 5. Write a program to input the marks of three subject (Out of 100) and calculate its percentage, total marks, total distinction in subjects and display fail if he/she gets marks less vthan 30 in any one of the subjects.
- 6. Write a program to input three numbers and find the greatest
 - (i) By using nested if statement
 - (ii) By using Ternary Operator
- 7. Write a program to input two numbers and perform addition, substraction and multiplication by using switchstatement.
- 8. Write a program to implement relational operator.(>,<,<=,=>,==,!=)
- 9. Write a program to implement bitwise operators.(&, |,^,~)
- 10. Write a program to implement shift operators. (<<,>>)
- 11. Write a program to display odd and even numbers between 5 to 50 and also display its sum separately.
- 12. Write a program to display and the sum of all numbers between 15 to 100, which is divisible by 7.
- 13. Write a program to input two numbers and swap them
 - (I) By using third variable.
 - (II) Without using third variable (Two methods)
- 14. Write a program to implement
 - the following series:-(1) 1 3 5 9
 - (2) 2 5 10 17 26 37
 - (3) 0 6 24 60 120 210
 - (4) 2 10 30 68 130 222
 - (5) 1 2 4 8 16 32
- 15. Write a program to create your own package and use it in your program.
- 16. Write a program to input a number and check it is prime or not.
- 17. Write a program to input a number and check it is palindrome or not.
- 18. Write a program to display all Armstrong numbers between 123 to 999.
- 19. Write a program to input a number and find all the prime number between 5 to 100

- 19. Write a program to input a decimal number and convert it into binary.
- 20. Write a program to input a binary number and convert it into decimal.
- 21. Write a program to input an octal number and convert it into decimal.

- 22. Write a program to input a decimal number and convert it into octal.
- 23. Write a program to input the elements of array and sort them by using selection sort.
- 24. Write a program to input a key element and search it from the given array.
- 25. Write a program to find the greatest and smallest element of the array.
- 26. Write a program to enter a number and find its factorial by using recursion.
- 27. Write a program to enter three numbers and find its GCD by using recursion.
- 28. Write a program to generate Fibonacci series by using recursion.
- 29. Write a program to input the elements of 3X3 matrix and display in matrix format.
- 30. Write a program to input the elements of 3X3 matrix and check it is symmetric or not.
- 31. Write a program to input the elements of 3X3 matrix and display the principal and reverse diagonal elements of the matrix.
- 32. Write a program to input the elements of 3X3 matrix and check it is sparse or dense.
- 33. Write a program to perform addition, substraction and multiplication of two matrix.
- 34. Write a program to implement all the pre-defined methods related to String .
- 35. Write a program to enter a string and perform:-
 - (1) the length of string
 - (2) reverse of string
 - (3) number of vowels
- 36. Write a program to implement default constructor.
- 37. Write a program to implement parametric constructor.
- 38. Write a program to implement constructor overloading.
- 39. Write a program to calculate factorial of any number by using default constructor.
- 40. Write a program to generate the following series by using default constructor
 - (1) 0 1 1 2 3 5 up to n terms.
 - (2) 2 3 5 8 13 21..... up to n terms.
- 41. Write a program to input n and r and implement permutation and combination.
- 42. Write a program to implement method overloading.
- 43. Write a program to implement single inheritance.
- 44. Write a program to implement multiple inheritances by using interface class.
- $\label{eq:45.} \ensuremath{\mathsf{Write}}\xspace a \ensuremath{\mathsf{program}}\xspace to implement multilevel inheritance.$
- 46. Write a program to implement hierarchal inheritance.
- 47. Write a program to implement abstract class.
- 48. Write a program to create your own package and use it in your program.
- 49. Write a program to implement exception handling:-
- By using Arithmetic Exception, Array Index Out of Bounds Exception, Number Format Exception and Input Mismatch Exception
- 50.Programs in HTML CSS
- ** Faculty may modify experiments as per syllabus requirements.

Course code: BC23-501

Category:

Course Title: Computer Networks	Semester: V
L-T-P : 3-1-0	Credit: 4

Detailed content

Unit 1:

Data communication Components: Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization:Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.

Unit 2:

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction -Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back –

N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA.

Unit 3:

Network Layer: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols.

Unit 4:

Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

Unit 5:

Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography.

Course code: BC23-502	Category:
Course Title: PYTHON PROGRAMMING	Semester: V
L-T-P : 3-1-0	Credit: 4

Course Objectives:

• To acquire programming skills in core Python.

- To acquire Object Oriented Skills in Python
- To develop the skill of designing Graphical user Interfaces in Python
- To develop the ability to write database applications in Python

Learning Outcomes:

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

- Explain basic principles of Python programming language
- Implement object oriented concepts,
- Implement database and GUI applications.

Unit – I

Introduction to Python, use IDLE to develop programs, Basic coding skills, working withdata types and variables, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure, Illustrative Programs.

Unit – II

Define and use functions and modules, working with recursion, Basic skills for working with lists, work with a list of lists, work with tuples, work with dates and times, get started with dictionaries, Illustrative programs.

Unit – III

An introduction to file I/O, use text files, use CSV files, use binary files, Handle a single exception, handle multiple exceptions, Illustrative programs.

Unit – IV

Object Oriented Programming, An introduction to classes and objects, define a class, work with object composition, work with encapsulation, work with inheritance, override object methods, Illustrative programs.

Unit –V

An introduction to relational databases, SQL statements for data manipulation, Using SQLite Manager to work with a database, Using Python to work with a database, Creating a GUI that handles an event, working with components, Illustrative programs.

Text Books and References:

- 1. Hands on Python, Kanahaiya Lal Ambashtha, The Foxland Publications, Published in 2021,Volume 1,1st edition
- 2. Programming Python: Powerful Object Oriented Programming; Mark Lutz; Shroff/O'Reilly; 2010.
- 3. Beginning Python: Using Python 2.6 & Python 3.1; James Payne; Wiley India; 2011.
- 4. Head First Programming: A Learner's Guide to Programming using Python Language; Barry & Griffiths; Shroff/O'Reilly; 2009.

Course Title: Design and Analysis of Algorithms Course code: BC23-503

LTP:3-1	LTP:3-1-0	
Credit:	4	
1	Introduction: Characteristics of algorithm.Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method andMasters' theorem	
2	Fundamental Algorithmic Strategies: Brute-Force, Greedy, Dynamic Programming, Branch and- Bound and Backtrackingmethodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving, Bin Packing, Knap Sack TSP. Heuristics –characteristics and their application domains.	
3	Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search(BFS); Shortest Path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm. Tractable and Intractable Problems: Computability.	
4	Design of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.	
6	Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE	

Text books/ reference books:

- Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, RonaldL Rivestand Clifford Stein, MIT Press/McGraw-Hill.
- Fundamentals of Algorithms E. Horowitz et al.
- Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
- Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
- Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House (AICTE Recommended Textbook 2018)
- Algorithms Design and Analysis, Udit Agarwal, Dhanpat Rai

Course Title: Computer Networks Laboratory Course code: BC23-591 LTP:0-0-4

Credit: 2

- 1. NIC Installation & Configuration (Windows/Linux).
- 2. Understanding IP address, subnet etc.

Familiarization with

- Networking cables (CAT5, UTP)
- Connectors (RJ45, T-connector)
- Hubs, Switches 3.

TCP/UDP 3. Socket

Programming

- Simple, TCP based, UDP based
- Multicast & Broadcast Sockets
- Implementation of a Prototype Multithreaded Server
- Implementation of Data Link Layer Flow Control Mechanism (Stop & Wait, Sliding Window) Data Link Layer Error Detection Mechanism (Cyclic Redundancy Check) Data Link Layer Error Control Mechanism (Selective Repeat, Go Back N)
- 1. Server Setup/Configuration : FTP, Telnet, NFS, DNS, Firewall

Course Title: Python Laboratory Course code: BC23-592 LTP:0-0-4 Credit: 2

Course Objectives:

- Master the fundamentals of writing Python scripts
- Learn core Python scripting elements such as variables and flow control structures
- Discover how to work with lists and sequence data
- Write Python functions to facilitate code reuse
- Use Python to read and write files
- Make their code robust by handling errors and exceptions properly
- Work with the Python standard library
- Explore Python's object-oriented features
- Search text using regular expressions

Learning Outcomes:

The students come to know to explain :-

- Analyse a problem and implement it using Python Programming.
- How an existing Python program works;
- How to find errors in a Python program and describe how to fix them;
- How Python improves other languages with object-oriented features.
- To familiarize the students with language environment.
- To implement various concepts related to language.
- To understand the difference between object oriented programming and procedural oriented language and data types in Python.
- To understand the GUI feature with different topics such as composition of objects,

Operatoroverloading, inheritance, Polymorphism etc.

- At the end of the course students will able to simulate the problem in the subjects like web pages, Computer networks and real world problems
- 1. Calculate Simple interest where principal amount, rate and time will be entered by user
- 2. Program to calculate the following:
 - a. Area of a triangle using hero's formula
 - b. Area of triangle in case height and base of triangle is given by user
 - c. Area and perimeter of rectangle
 - d. Area and perimeter of square
 - e. Area and perimeter of trapezium
 - f. Area and perimeter of parallelogram
 - g. Area and perimeter of circle
 - h. Area and volume of cube
 - i. Area and volume of cuboid
 - j. Area and volume of sphere
- 3. Program to check largest number among three numbers
- 4. Program to solve quadratic equation
- 5. Program to convert Celsius to Fahrenheit
- 6. Program to convert Fahrenheit to Celsius
- 7. Program to input marks of 5 subjects and calculate percentage and division obtained by a student
- 8. Program to check whether inputted number is even number or odd number.
- 9. Program to input a year and check is it a leap year or not?
- 10. Program to check an entered number is Armstrong or not?
- 11. Program to print all the even number between the given range.
- 12. Program to print all the odd number between the given range.
- 13. Program to calculate the sum of all natural numbers between the given interval.
- 14. Program to print all the Armstrong number between the given range.
- 15. Program to check a number is prime or not.
- 16. Program to print all the prime number between the given range.
- 17. Program to check a number is palindrome or not?
- 18. Program to print the reverse of the given number
- 19. Program to print Fibonacci series up to n terms.
- 20. Program to find the factorial of the number
- 21. Program to display the multiplication table of a given number.
- 22. Program to find LCM

Program Based on Function:

- 1. Program to convert Celsius to Fahrenheit using function
- 2. Program to find factorial of entered number using function.
- 3. Program to check a number is palindrome or not using function
- 4. Program to print the reverse of the given number using function

- 5. Program to print prime no in the given interval using function. Interval must be entered by user.
- 6. Program to find factorial of entered number using recursive function
- 7. Program to print Fibonacci series using recursive function
- 8. Program to find the sum of n natural number using recursive function
- 9. Program to implement different arithmetic operation using modular approach
- 10. Program to implement conversion of different units into another using modular approach

Program Based on File Handling:

- **1.** Python program to create a text file, write the content of the file from console and print its contenton the console.
- **2.** Python program to append an existing file and display the contents of the updated file on thescreen.
- **3.** Python program to copy the content of a file into another file.
- 4. Python program to concatenate two files into third file.
- 5. Python program to rename the file.

Program based on exceptional handling:

- 1. Python program to handle "Zero Division Error" exception
- 2. Python program to handle "Value Error" exception
- 3. Python program to handle "File Not Found" exception
- **4.** Python program to handle "Index Error" exception

Program Based on Class

- 1. Create a class student. Define a function to input a name and display the name through object.
- 2. Create a class Employee. Define a function to take employee data as input and display the data

Program Based on GUI

- 1. Design an interface to take input of two numbers in different text box and by click on ADD button, display the sum of those entered number.
- 2. Design an interface to take input of your name. Display the name on click of DISPLAY button.

Course Title: Environment and Ecology Course code:BC23-601 LTP: 3-1-0 Credit: 4 Introduction, components of the environment, environmental degradation

Ecology: Elements of Ecology ; Ecological balance and consequences of change, principles of environmental impact assessment

Air Pollution and Control: Atmospheric composition, energy balance, climate, weather, dispersion, sources and effects of pollutants, primary and secondary pollutants, green house effect, depletion of ozone layer, standards and control measures.

Water Pollution and Control: Hydrosphere, natural water, pollutants: their origin and effects, river / lake / ground water pollution, standards and control.

Land Pollution: Lithosphere, pollutants (municipal, industrial, commercial, agricultural, hazardous solid wastes); their origin and effects, collection and disposal of solid waste, recovery and conversion methods.

Noise Pollution: Sources, effects, standards and control.

Books:

- Environmental Science, Cunningham, TMH
- Environmental Pollution Control Engineering, C.S. Rao, New Age International
- Environmental Science, Wright & Nebel, PHI
- Environmental Pollution Analysis, S.M.Khopkar, New Age International
- Environmental Mgmt, N.K. Oberoi, EXCEL
- Environmental Mgmt, Mukherjee, VIKAS
- Ecosystem Principles & Sustainable Agriculture, sithambaranathan, Scitech

Course Title: Image Processing Course code:BC23-E501A LTP: 3-1-0

Unit	Content
1	Introduction
	Background, Digital Image Representation, Fundamental steps in Image
	Processing, Elements of Digital Image Processing - Image Acquisition, Storage,
	Processing, Communication, Display.
	Digital Image Formation
2	A Simple Image Model, Geometric Model- Basic Transformation (Translation,
	Scaling, Rotation), Perspective Projection, Sampling & Quantization
	-Uniform & Non uniform.
	Mathematical Preliminaries
3	Neighbourofpixels,Connectivity,Relations, Equivalence & Transitive
	Closure; Distance Measures, Arithmetic/LogicOperations, Fourier
	Transformation, Properties of The Two Dimensional Fourier Transform,
	Discrete Fourier Transform, Discrete Cosine & Sine Transform.
4.	Image EnhancementSpatial Domain Method, Frequency Domain Method, Contrast Enhancement -Linear & NonlinearStretching,Histogram Processing;Smoothing - Image Averaging, Mean Filter, Low-passFiltering;ImageSharpening. High- pass Filtering, High- boost Filtering, Derivative Filtering,Homomorphic Filtering;Enhancement in the frequency domain - Low passfiltering, High pass filtering.
5	Image RestorationDegradation Model, Discrete Formulation, Algebraic Approach to RestorationUnconstrained& Constrained; Constrained LeastSquareRestoration, Restoration by Homomorphic Filtering, Geometric TransformationSpatial Transformation, Gray LevelInterpolation.
6	Image Segmentation Point Detection Line Detection Edge detection Combined detection
	Edge Linking & Boundary Detection - Local Processing Global
	Processing via The Hough Transform: Thresholding - Foundation Simple
	Global Thresholding Ontimal Thresholding Region Oriented
	Segmentation - Basic Formulation Region Growing by Divel Aggregation
	segmentation basic of matation, region drowing by the Aggregation,

TEXT BOOKS:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson

Education, 6th Edition.

2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition.

REFERENCE BOOKS:

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.

2. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition.

3. Information Security, Principles, and Practice: Mark Stamp, Wiley India.

4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH.

5. Introduction to Network Security: Neal Krawetz, CENGAGE Learning.

6. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

Course Title: Pattern Recognition Course code:BC23-E501B LTP: 3-1-0 Credit: 4

Unit	Content
	1. Basics of pattern recognition
	2. Bayesian decision theory Classifiers, Discriminant functions, Decision surfaces Normal density and discriminant functions Discrete features
	3. Parameter estimationmethods Maximum-Likelihood estimation Gaussian mixture models Expectation- maximization method Bayesian estimation
	4. Hidden Markov models for sequential pattern classification Discrete hidden Markov models Continuous density hidden Markov models
	 Dimension reduction methods Fisher discriminant analysis, Principal component analysis, Parzen-window method, K-Nearest Neighbour method
	6. Non-parametric estimation techniques for Density estimation
	7. Linear discriminant function based classifier Perceptron Support vector machines
	8. Non-metric methods for pattern classification Non-numeric data or nominal data Decision trees
	9. Unsupervised learning and clustering Criterion functions for clustering Algorithms for clustering: K-means, Hierarchica and other methods

Course Title: Compiler Design

Course code:BC23-E501C LTP: 3-1-0 Credit: 4

Objective:

- 1. To understand and list the different stages in the process of compilation.
- 2. Identify different methods of lexical analysis
- 3. Design top-down and bottom-up parsers
- 4. Identify synthesized and inherited attributes
- 5. Develop syntax directed translation schemes
- 6. Develop algorithms to generate code for a target machine

Unit 1: Introduction to Compiling

Compilers, Analysis of the source program, The phases of the compiler, Cousins of the compiler.

Unit 2: Lexical Analysis

Theroleofthelexical analyzer, Tokens, Patterns, Lexemes, Inputbuffering, Specifications of a token, Recognition of a tokens, Finite automata, from a regular expression to an NFA, From a regular expression to NFA, From a regular expression to DFA, Design of a lexical analyzer generator (Lex).

Unit 3: Syntax Analysis

The role of a parser, Context freegrammars, Writing a grammar, Top downParsing, Non- recursive Predictive parsing (LL), Bottom up parsing, Handles, Viable prefixes, Operator precedence parsing, LR parsers (SLR, LALR), Parsergenerators (YACC). Error Recovery strategies for different parsing techniques.

Unit 4: Syntax directed translation

Syntax director definitions, Construction of syntaxtrees, Bottom-up evaluation of S attributed definitions, L attributed definitions, Bottom-up evaluation of inherited attributes.

Unit 5: Type checking

Typesystems, Specification of a simple type checker, Equivalence of type expressions, Type conversions.

Unit 6: Run time environments

Sourcelanguageissues (Activation trees, Control stack, scope of declaration, Binding of names), Storage organization (Subdivision of run-time memory, Activation records), Storage allocation strategies, Parameter passing (call by value, call by reference, copy restore, call by name), Symbol tables, dynamic storage allocation techniques

Unit 7: Intermediate code generation

Intermediate languages, Graphical representation, Three-address code, Implementation of three address statements (Quadruples, Triples, Indirect triples).

Unit 8: Code optimization

Introduction, Basic blocks & flow graphs, Transformation of basic blocks, Dag representation of basic blocks, The principle sources of optimization, Loops in

flow graph, Peephole optimization.

Unit 9: Code generations

Issues in the design of code generator, a simple code generator, Register allocation & assignment.

Text book and Reference books:

Aho, Sethi, Ullman - "Compiler Principles, Techniques and Tools" - Pearson Education.
 Holub - "Compiler Design in C" - PHI.

Course Outcomes:

- On completion of the course students will be able to
- Understand given grammar specification develop the lexical analyser
- Design a given parser specification design top- do and bottom-up parsers
- Develop syntax directed translation schemes
- Develop algorithms to generate code for a target machine