KAZI NAZRUL UNIVERSITY | **FACULTY OF SCIENCE** | **DEGREE PROGRAMME: B.Sc. HONOURS IN COMPUTER SCIENCE** | **WITH EFFECT FROM THE ACADEMIC SESSION: 2020-21**

Abbreviated Degree	Discipline	Abbreviated Degree Programme	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
												Practical	Theoretical	Practical	Theoretical		
BSCH	**COMPUTER SCIENCE**	**BSCHCOS**	**I**	Programming Methodology		C	BSCHCOSC101	CC-1	4 - 0 - 4	6	22	30	10	20	40	100	250 + Marks of GEC
				Computer System Architecture			BSCHCOSC102	CC-2	4 - 0 - 4	6		30	10	20	40	100	
				Choose from the Pool of Generic Elective Courses offered in 1st Semester by Honours Disciplines other than the Discipline in which Honours course is taken		GE	See Pool	GEC-1	See Pool	6		See Pool				Marks of GEC	
				Environmental Studies		AE	AEE101	AECC-1	4 - 0 - 0	4			10		40	50	
			II	Data Structure		C	BSCHCOSC201	CC-3	4 - 0 - 4	6	22	30	10	20	40	100	200 + Marks of GEC
				Discrete Structures			BSCHCOSC202	CC-4	5 - 1 - 0	6			10		40	50	
				Choose from the Pool of Generic Elective Courses offered in 2nd Semester by Honours Disciplines other than the Discipline in which Honours course is taken		GE	See Pool	GEC-2	See Pool	6		See Pool				Marks of GEC	
				English/MIL Communication		AE	See Pool	AECC-2	4 - 0 - 0	4			10		40	50	
			III	Operating System		C	BSCHCOSC301	CC-5	4 - 0 - 4	6	28	30	10	20	40	100	300 + Marks of GEC
				Analysis of Algorithms			BSCHCOSC302	CC-6	5 - 1 - 0	6			10		40	50	
				Computer Networks			BSCHCOSC303	CC-7	4 - 0 - 4	6		30	10	20	40	100	
				Choose from the Pool of Generic Elective Courses offered in 3rd Semester by Honours Disciplines other than the Discipline in which Honours course is taken		GE	See Pool	GEC-3	See Pool	6		See Pool				Marks of GEC	
				Programming in Java	(Any One)	SE	BSCHCOSSE301	SEC-1	0 - 0 - 8	4		30		20		50	
				Python Programming			BSCHCOSSE302		0 - 0 - 8			30		20			
			IV	Software Engineering		C	BSCHCOSC401	CC-8	5 - 1 - 0	6	28		10		40	50	300 + Marks of GEC
				Database Management System			BSCHCOSC402	CC-9	4 - 0 - 4	6		30	10	20	40	100	
				Object Oriented Programming			BSCHCOSC403	CC-10	4 - 0 - 4	6		30	10	20	40	100	
				Choose from the Pool of Generic Elective Courses offered in 4th Semester by Honours Disciplines other than the Discipline in which Honours course is taken		GE	See Pool	GEC-4	See Pool	6		See Pool				Marks of GEC	
				Mobile Application Development	(Any One)	SE	BSCHCOSSE401	SEC-2	0 - 0 - 8	4		30		20		50	
				Web Programming			BSCHCOSSE402		0 - 0 - 8			30		20			
			V	Internet Technologies		C	BSCHCOSC501	CC-11	4 - 0 - 4	6	24	30	10	20	40	100	300
				Artificial Intelligence			BSCHCOSC502	CC-12	4 - 0 - 4	6		30	10	20	40	100	
				Image Processing	(Any Two)	DSE	BSCHCOSDSE501	DSEC-1 & DSEC-2	5 - 1 - 0	6 X 2			10		40	50 X 2	
				Data Analytics			BSCHCOSDSE502		5 - 1 - 0				10		40		
				Computer Ethics			BSCHCOSDSE503		5 - 1 - 0				10		40		
				System Security			BSCHCOSDSE504		5 - 1 - 0				10		40		
				Human Computer Interface			BSCHCOSDSE505		5 - 1 - 0				10		40		
			VI	Computer Graphics		C	BSCHCOSC601	CC-13	4 - 0 - 4	6	24	30	10	20	40	100	300
				Machine Learning			BSCHCOSC602	CC-14	4 - 0 - 4	6		30	10	20	40	100	
				Modelling and Simulation	(Any Two)	DSE	BSCHCOSDSE601	DSEC-3 & DSEC-4	5 - 1 - 0	6 X 2			10		40	50 X 2	
				Theory of Computation			BSCHCOSDSE602		5 - 1 - 0				10		40		
				Data Mining			BSCHCOSDSE603		5 - 1 - 0				10		40		
				Cloud Computing			BSCHCOSDSE604		5 - 1 - 0				10		40		
				Internet of Things			BSCHCOSDSE605		5 - 1 - 0				10		40		
				Total Credit and Marks							**148**						**1650 + Marks of GEC**

Abbreviations: C= Core; CC=Core Course; AE= Ability Enhancement; AECC= Ability Enhancement Compulsory Course; GE= Generic Elective; GEC= Generic Elective Course; SE= Skill Enhancement; SEC= Skill Enhancement Course; DSE= Discipline Specific Elective; DSEC= Discipline Specific Elective Course; CA= Continuous Assessment, ESE= End Semester Examination, L= Lecture Hour; T= Tutorial Hour and P= Practical Hour/ Field Work and NA= Not Applicable

Generic Elective Courses (GEC) : Students of a particular Honours Course will choose from the Pool of Generic Elective Courses offered by Disciplines other than the Discipline in which Honours course is taken

Semesterwise Pool of Generic Elective Courses offered by this Discipline

Discipline	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
										Practical	Theoretical	Practical	Theoretical		
COMPUTER SCIENCE	**I**	Programming Methodology		GE	BSCHCOSGE101	GEC-1	4 - 0 - 4	6	NA	30	10	20	40	100	NA
	II	Data Structure			BSCHCOSGE201	GEC-2	4 - 0 - 4	6		30	10	20	40	100	
	III	Operating System and Shell Scripts	(Any One)		BSCHCOSGE301	GEC-3	4 - 0 - 4	6		30	10	20	40	100	
		Introduction to Computer Networks			BSCHCOSGE302		4 - 0 - 4			30	10	20	40		
	IV	Object Oriented Programming in C++	(Any One)		BSCHCOSGE401	GEC-4	4 - 0 - 4	6		30	10	20	40	100	
		Introduction to Database Management System			BSCHCOSGE402		4 - 0 - 4			30	10	20	40		

Pool of Communication Courses offered as Ability Enhancement Compulsary Courses

Discipline	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
										Practical	Theoretical	Practical	Theoretical		
English/MIL Communication	**II**	English Communication	(Any One)	AE	AECCE201	AECC-2	4 - 0 - 0	4	NA		10		40	50	NA
		Bengali Communication			AECCB201		4 - 0 - 0				10		40		
		Hindi Communication			AECCH201		4 - 0 - 0				10		40		
		Urdu Communication			AECCU201		4 - 0 - 0				10		40		

KAZI NAZRUL UNIVERSITY | **FACULTY OF SCIENCE** | **DEGREE PROGRAMME: B.Sc. PROGRAM IN COMPUTER SCIENCE** | **WITH EFFECT FROM THE ACADEMIC SESSION: 2020-21**

Abbreviated Degree	Discipline	Abbreviated Degree Programme	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
												Practical	Theoretical	Practical	Theoretical		
BSCP	COMPUTER SCIENCE	BSCPCOS	I	Programming Methodologies		C	BSCPCOSC101	CC-1(1)	4 - 0 - 4	6	22	30	10	20	40	100	150 + Marks of Disc.2 & Disc.3 (CC)
				Discipline 2: CC-1(1) of Other Discipline			See Pool	CC-2(1)	See Pool	6		See Pool				Marks of Disc.2 & Disc.3	
				Discipline 3: CC-1(1) of Other Discipline				CC-3(1)		6							
				Environment Studies		AE	AEE101	AECC-1	4 - 0 - 0	4			10		40	50	
			II	Data Structure		C	BSCPCOSC201	CC-1(2)	4 - 0 - 4	6	22	30	10	20	40	100	150 + Marks of Disc.2 & Disc.3 (CC)
				Discipline 2: CC-1(2) of Other Discipline			See Pool	CC-2(2)	See Pool	6		See Pool				Marks of Disc.2 & Disc.3	
				Discipline 3: CC-1(2) of Other Discipline				CC-3(2)		6							
				English/MIL Communication		AE	See Pool	AECC-2	4 - 0 - 0	4			10		40	50	
			III	Basics of Operating Systems		C	BSCPCOSC301	CC-1(3)	5 - 1 - 0	6	22		10		40	50	100 + Marks of Disc.2 & Disc.3 (CC)
				Discipline 2: CC-1(3) of Other Discipline			See Pool	CC-2(3)	See Pool	6		See Pool				Marks of Disc.2 & Disc.3	
				Discipline 3: CC-1(3) of Other Discipline				CC-3(3)		6							
				Core Java		SE	BSCPCOSSE301	SEC-1	0 - 0 - 8	4		30		20		50	
			IV	Basics of Database Management System		C	BSCPCOSC401	CC-1(4)	4 - 0 - 4	6	22	30	10	20	40	100	150 + Marks of Disc.2 & Disc.3 (CC)
				Discipline 2: CC-1(4) of Other Discipline			See Pool	CC-2(4)	See Pool	6		See Pool				Marks of Disc.2 & Disc.3	
				Discipline 3: CC-1(4) of Other Discipline				CC-3(4)		6							
				Web Programming with PHP		SE	BSCPCOSSE401	SEC-2	0 - 0 - 8	4		30		20		50	
			V	Basics of Software Engineering	(Any One)	DSE	BSCPCOSDSE501	DSEC-1(1)	5 - 1 - 0	6	22		10		40	50	100 + Marks of Disc.2 & Disc.3 (DESC)
				Basics of Computer Networks			BSCPCOSDSE502		5 - 1 - 0				10		40		
				Discipline 2: DSE-1(1) of Other Discipline			See Pool	DSEC-2(1)	See Pool	6		See Pool				Marks of DSE chosen	
				Discipline 3: DSE-1(1) of Other Discipline				DSEC-3(1)		6							
				MATLAB Programming	(Any One)	SE	BSCPCOSSE501	SEC-3	0 - 0 - 8	4		30		20		50	
				Programming in Python			BSCPCOSSE502		0 - 0 - 8			30		20			
			VI	Basics of Artificial Intelligence	(Any One)	DSE	BSCPCOSDSE601	DSEC-1(2)	4 - 0 - 4	6	22	30	10	20	40	100	150 + Marks of Disc.2 & Disc.3 (DESC)
				Basics of Computer Graphics			BSCPCOSDSE602		4 - 0 - 4			30	10	20	40		
				Discipline 2: DSE-1(2) of Other Discipline			See Pool	DSEC-2(2)	See Pool	6		See Pool				Marks of DSE chosen	
				Discipline 3: DSE-1(2) of Other Discipline				DSEC-3(2)		6							
				Basics of Mobile Application Development	(Any One)	SE	BSCPCOSSE601	SEC-4	4 - 0 - 0	4			10		40	50	
				Basics of Cloud Computing			BSCPCOSSE602		4 - 0 - 0				10		40		
				Total Credit and Marks							**132**						**800 + Marks of Disc.2 & Disc.3 CC and DSEC**

Abbreviations: C= Core; CC=Core Course; AE= Ability Enhancement; AECC= Ability Enhancement Compulsory Course; GE= Generic Elective; GEC= Generic Elective Course; SE= Skill Enhancement; SEC= Skill Enhancement Course; DSE= Discipline Specific Elective; DSEC= Discipline Specific Elective Course; CA= Continuous Assessment, ESE= End Semester Examination, L= Lecture Hour; T= Tutorial Hour and P= Practical Hour/ Field Work and NA= Not Applicable

Discipline-2: Students of a particular Program Course will choose Discipline-2 of any other Discipline except of his/her own. For example, if a student opts for Program Course in Computer Science then that student is required to opt Discipline-2 from Program Courses other than Computer Science. Once Discipline-2 is chosen in the 1st Semester the student is to continue with the same in the remaining semesters.

Discipline-3: Students of a particular Program Course will choose Discipline-3 of any other Discipline except Discipline 1 and Discipline 2 already chosen. Once Discipline-3 is chosen in the 1st Semester the student is to continue with the same in the remaining semesters.

Semesterwise Pool of Core Courses offered by this Discipline (when treated as Discipline 2 for other Program Courses across the faculties as far as practicable)

Discipline-2	Semester	Course Name	Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
									Practical	Theoretical	Practical	Theoretical		
COMPUTER SCIENCE	I	Programming Methodologies	C	BSCPCOSC101	CC-2(1)	4 - 0 - 4	6	NA	30	10	20	40	100	NA
	II	Data Structure		BSCPCOSC201	CC-2(2)	4 - 0 - 4	6		30	10	20	40	100	
	III	Basics of Operating Systems		BSCPCOSC301	CC-2(3)	4 - 0 - 4	6		30	10	20	40	100	
	IV	Basics of Database Management System		BSCPCOSC401	CC-2(4)	4 - 0 - 4	6		30	10	20	40	100	

Semesterwise Pool of Core Courses offered by this Discipline (when treated as Discipline 3 for other Program Courses across the facultiesas far as practicable)

Discipline-3	Semester	Course Name	Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
									Practical	Theoretical	Practical	Theoretical		
COMPUTER SCIENCE	I	Programming Methodologies	C	BSCPCOSC101	CC-3(1)	4 - 0 - 4	6	NA	30	10	20	40	100	NA
	II	Data Structure		BSCPCOSC201	CC-3(2)	4 - 0 - 4	6		30	10	20	40	100	
	III	Basics of Operating Systems		BSCPCOSC301	CC-3(3)	4 - 0 - 4	6		30	10	20	40	100	

	IV	Basics of Database Management System			BSCPCOSC401	CC-3(4)	4 - 0 - 4	6		30	10	20	40	100	

Discipline Specific Elective Course (DSEC) -2: Students of a particular Program Course will choose DSEC-2 from DSEC-1 of Discipline 2.

Discipline Specific Elective Course (DSEC) -3: Students of a particular Program Course will choose DSEC-3 from DSEC-1 of Discipline 3.

Semesterwise Pool of Discipline Specific Elective Courses (DSEC) offered by this Discipline (when treated as Discipline 2 for other Program Courses across the faculties as far as practicable)

Discipline-2	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
										Practical	Theoretical	Practical	Theoretical		
COMPUTER SCIENCE	**V**	Basics of Software Engineering	(Any One)	DSE	BSCPCOSDSE501	DSEC-2(1)	5 - 1 - 0	6	NA		10		40	50	NA
		Basics of Computer Networks			BSCPCOSDSE502		5 - 1 - 0				10		40		
	VI	Basics of Artificial Intelligence	(Any One)		BSCPCOSDSE601	DSEC-2(2)	4 - 0 - 4	6		30	10	20	40	100	
		Basics of Computer Graphics			BSCPCOSDSE602		4 - 0 - 4			30	10	20	40		

Semesterwise Pool of Discipline Specific Elective Courses (DSEC) offered by this Discipline (when treated as Discipline 3 for other Program Courses across the faculties as far as practicable)

Discipline-3	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
										Practical	Theoretical	Practical	Theoretical		
COMPUTER SCIENCE	**V**	Basics of Software Engineering	(Any One)	DSE	BSCPCOSDSE501	DSEC-3(1)	4 - 0 - 4	6	NA		10		40	50	NA
		Basics of Computer Networks			BSCPCOSDSE502		4 - 0 - 4				10		40		
	VI	Basics of Artificial Intelligence	(Any One)		BSCPCOSDSE601	DSEC-3(2)	4 - 0 - 4	6		30	10	20	40	100	
		Basics of Computer Graphics			BSCPCOSDSE602		4 - 0 - 4			30	10	20	40		

Pool of Communication Courses offered as Ability Enhancement Compulsary Courses

Discipline	Semester	Course Name		Course Type	Course Code	Course Details	L - T - P	Course Credit	Sem Credit	CA Marks		ESE Marks		Total Marks	Sem Marks
										Practical	Theoretical	Practical	Theoretical		
English/MIL Communication	**II**	English Communication	(Any One)	AE	AECCE201	AECC-2	4 - 0 - 0	4	NA		10		40	50	NA
		Bengali Communication			AECCB201		4 - 0 - 0				10		40		
		Hindi Communication			AECCH201		4 - 0 - 0				10		40		
		Urdu Communication			AECCU201		4 - 0 - 0				10		40		

Learning Outcome based Curriculum Framework (LOCF)

For

Choice Based Credit System (CBCS)

Syllabus

B.Sc. (Honours) in Computer Science

w.e.f. Academic Session 2020-21

Kazi Nazrul University

Asansol, Paschim Bardhaman
West Bengal 713340

PREAMBLE ACCORDING TO UGC LOCF to be included

Semester- I

Course Name: PROGRAMMING METHODOLOGY

Course Code: BSCHCOSC101

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-1**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn to develop simple algorithms and flow charts to solve a problem.*
2. *Develop problem solving skills coupled with top down design principles.*
3. *Learn about the strategies of writing efficient and well-structured computer algorithms/programs.*
4. *Develop the skills for formulating iterative solutions to a problem.*
5. *Learn array processing algorithms coupled with iterative methods.*
6. *Learn text and string processing efficient algorithms.*
7. *Learn searching techniques and use of pointers.*
8. *Understand recursive techniques in programming.*

Course Content:

Theory

UNIT I. Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies, Introduction to C/C++ Programming - Basic Program Structure In C/C++, Variables and Assignments, Input and Output, Selection and Repetition Statements.

UNIT II. Top-Down Design, Predefined Functions, Programmer -defined Function, Local Variable, Functions with Default Arguments

UNIT III. Introduction to Arrays, Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

UNIT IV. Pointers - Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Call-By-Value and Call-By-Reference Parameters.

UNIT V. Structures - Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures, Unions.

UNIT VI. Strings - Declaration and Initialization, Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions.

UNIT VII. Searching Algorithms - Linear Search, Binary Search. Use of files for data input and output. merging and copy files.

UNIT VIII. Recursion - Developing Recursive Definition of Simple Problems and their implementation.

Practical

UNIT I. Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code, execute and test it. Students should be given assignments on following :

a) To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures

b) Learn how to use functions and parameter passing in functions, writing recursive programs.

UNIT II. Students should be given assignments on following:

a) Write Programs to learn the use of strings and string handling operations.

b) Problems which can effectively demonstrate use of Arrays. Structures and Union.

c) Write programs using pointers.

d) Write programs to use files for data input and output.

e) Write programs to implement search algorithms.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015.
2. Programming and problem solving with C++: brief edition, N. Dale and C. Weems, Jones & Bartlett Learning, 2010.
3. C Programming, Karnighan,&Ritchie, PHI
4. Programming through C, Richard Johnsonbaugh and Martin Kalin, Pearson Education
5. Programming in C, B.S. Gottfried, Sahaum Series.
6. Programming in ANSI C, E. Balaguruswami, TMH

Course Name: Computer System Architecture

Course Code: BSCHCOSC102

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-2**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. To make students understand the basic structure, operation and characteristics of digital computer.
2. To familiarize the students with arithmetic and logic unit as well as the concept of the concept of pipelining.
3. To familiarize the students with hierarchical memory system including cache memories and virtual memory.
4. To make students know the different ways of communicating with I/O devices and standard I/O interfaces.

Course Content:

Theory

UNIT I. Introduction - Logic gates (OR, AND, NOT, NAND, NOR, Exclusive – OR, Exclusive – NOR, Mixed logic), Boolean algebra, Map Simplification (Sum of Products, Product of Sums, Karnaugh Map (up to 4 variables)), combinational circuits (Adder, Subtractor, Multiplexer, Demultiplexer, Decoder, Encoder, Comparator, Code converter), Circuit simplification, flip-flops (SR,T,D,JK,MASTER SLAVE) and sequential circuits (registers, counters and memory units).

UNIT II. Data Representation and Basic Computer Arithmetic - Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers.

UNIT III. Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus & Memory Transfer, Arithmetic Microoperations, Logic Microoperations, Shift Microoperation.

UNIT IV. Basic Computer Organization: Instruction codes, Computer Registers, Computer Instructions, Timing & Control, Instruction Cycles, Memory Reference Instruction, Input - Output & Interrupts, Complete Computer Description & Design of Basic Computer.

UNIT V. Processor and Control Unit: Hardwired vs. Micro programmed Control Unit, General Register Organization, Stack Organization, Instruction Format, Data Transfer & Manipulation, Program Control, RISC, CISC, Pipelining – Pipelined datapath and control – Handling Data hazards & Control hazards. Introduction to Parallelism.

UNIT VI. Memory and I/O Systems: Peripheral Devices, I/O Interface, Data Transfer Schemes, Program Control, Interrupt, DMA Transfer, I/O Processor. Memory Hierarchy, Processor vs. Memory Speed, High-Speed Memories, Cache Memory, Associative Memory, Interleave, Virtual Memory, Cache Mapping Techniques, Memory Management.

Practical

UNIT I. Design of combinational circuits.

UNIT II. Design of sequential circuits.

UNIT III. Create a fetch routine of the instruction cycle.

UNIT IV. Create a machine based on the given architecture (Register Sets, Memory, Instruction format and basic computer instructions).

Internal (CA) Evaluation: Practical Note Book (15 marks), One experiment (10 marks) – from Unit III or Unit IV, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Unit I and another from Unit II, Viva-voce (10 marks)

References/ Suggested Readings:

1. Digital Logic and Computer Design, M. Morris Mao, PHI.
2. Computer System Architecture, M. Morris Mano, 3rd Edition, Prentice Hall.
3. Computer Organization and Design, David A. Patterson and John L. Hennessey, Fifth edition, Morgan Kauffman / Elsevier, 2014.
4. Floyd, Digital Fundamentals, Pearson Education.
5. Computer Architecture: A Quantitative Approach, John L. Hennessy, David A. Patterson, 4th Edition.
6. Computer Organization and Architecture, William Stallings, Prentice Hall.

Course Name: PROGRAMMING METHODOLOGY

Course Code: BSCHCOSGE101

<table>
<tr><td>Course Type: <b>GE (Theoretical & Practical)</b></td><td colspan="3">Course Details: <b>GEC-1</b></td><td colspan="2">L-T-P: <b>4 - 0 - 4</b></td></tr>
<tr><td rowspan="3">Credit: <b>6</b></td><td rowspan="3">Full Marks: <b>100</b></td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td><b>30</b></td><td><b>10</b></td><td><b>20</b></td><td><b>40</b></td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn to develop simple algorithms and flow charts to solve a problem.*
2. *Develop problem solving skills coupled with top down design principles.*
3. *Learn about the strategies of writing efficient and well-structured computer algorithms/programs.*
4. *Develop the skills for formulating iterative solutions to a problem.*
5. *Learn array processing algorithms coupled with iterative methods.*
6. *Learn text and string processing efficient algorithms.*
7. *Learn searching techniques and use of pointers.*
8. *Understand recursive techniques in programming.*

Course Content:

Theory

UNIT I. Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies, Introduction to C/C++ Programming - Basic Program Structure In C/C++, Variables and Assignments, Input and Output, Selection and Repetition Statements.

UNIT II. Top-Down Design, Predefined Functions, Programmer -defined Function, Local Variable, Functions with Default Arguments

UNIT III. Introduction to Arrays, Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

UNIT IV. Pointers - Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Call-By-Value and Call-By-Reference Parameters.

UNIT V. Structures - Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures, Unions.

UNIT VI. Strings - Declaration and Initialization, Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions.

UNIT VII. Searching Algorithms - Linear Search, Binary Search. Use of files for data input and output. merging and copy files.

UNIT VIII. Recursion - Developing Recursive Definition of Simple Problems and their implementation.

Practical

UNIT I. Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code, execute and test it. Students should be given assignments on following:

a. To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures

b. Learn how to use functions and parameter passing in functions, writing recursive programs.

UNIT II. Students should be given assignments on following:

a. Write Programs to learn the use of strings and string handling operations.

b. Problems which can effectively demonstrate use of Arrays. Structures and Union.

c. Write programs using pointers.

d. Write programs to use files for data input and output.

e. Write programs to implement search algorithms.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015.
2. Programming and problem solving with C++: brief edition, N. Dale and C. Weems, Jones & Bartlett Learning, 2010.
3. C Programming, Karnighan,&Ritchie, PHI
4. Programming through C, Richard Johnsonbaugh and Martin Kalin, Pearson Education
5. Programming in C, B.S. Gottfried, Sahaum Series.
6. Programming in ANSI C, E. Balaguruswami, TMH

Semester- II

Course Name: DATA STRUCTURE

Course Code: BSCHCOSC201

<table>
<tr><td>Course Type: <b>Core (Theoretical & Practical)</b></td><td colspan="3">Course Details: <b>CC-3</b></td><td colspan="2">L-T-P: <b>4 - 0 - 4</b></td></tr>
<tr><td rowspan="3">Credit: <b>6</b></td><td rowspan="3">Full Marks: <b>100</b></td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td><b>30</b></td><td><b>10</b></td><td><b>20</b></td><td><b>40</b></td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles*
2. *To have knowledge of complexity of basic operations like insert, delete, search on these data structures.*
3. *Ability to choose a data structure to suitably model any data used in computer applications.*
4. *Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.*
5. *Ability to assess efficiency tradeoffs among different data structure implementations.*
6. *Implement and know the applications of algorithms for sorting, searching etc.*

Course Content:

Theory

UNIT I. Basic concepts- Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis, Linear and Non Linear data structures, Singly Linked Lists-Operations, Concatenating, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

UNIT II. Stack- Definition and Operations, Array and Linked Implementations, Applications - Valid Expression Checking (Parenthesis matching), Reversal of string, Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation.

UNIT III. Queue - Definition and Operations, Array and Linked Implementations, Applications, Circular Queues - Insertion and Deletion Operations, Dequeue (Double Ended Queue) - Introduction.

UNIT IV. Sorting Methods – Bubble, Insertion, Selection, Shell, Using Divide-Conquer Approach (Quick and Merge sort), Comparison of Sorting Methods, Searching Methods – Linear and Binary.

UNIT V. Trees, Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees, Binary Search tree - Creation, Insertion, Deletion and Search, AVL tree-Definition, Examples, Insertion and Rotations, B tree, B+ tree, Priority Queue- Definition and Implementation, Heap- Definition, Min heap, Max heap, Insertion and Deletion.

UNIT VI. Static Hashing- Introduction, Hash tables, Hash functions, Overflow Handling.

Practical

Students are required to write and practically execute programs to solve problem using various data structures. The teacher can suitably device problems which help students experiment using the suitable data structures and operations. Some of the problems are indicated below.

1. Write program that uses functions to perform the following:
   a) Creation of list of elements where the size of the list, elements to be inserted and deleted are dynamically given as input.
   b) Implement the operations, insertion, deletion at a given position in the list and search for an element in the list
   c) To display the elements in forward / reverse order
2. Write a program to implement stack data structure and basic operations on it (Insertion, deletion). Write a program that demonstrates the application of stack operations (Eg: infix expression to postfix conversion, postfix evaluation).
3. Write a program to implement queue data structure and basic operations on it (Insertion, deletion, find length) and code at least one application using queues.
4. Write program that implements linear and binary search methods of searching for an elements in a list.
5. Write and trace programs to understand the various phases of sorting elements using the methods a) Bubble sort b) Insertion Sort c) Quicksort etc.
6. Write a program to create a Binary search tree and insert and delete from the tree. Write recursive and non-recursive routines to traverse a binary tree in preorder, inorder and postorder.
7. Write programs for recursion (Eg. Fibonacci numbers, Towers of Hanoi).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. Fundamentals of Data structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson-Freed, Universities Press.

2. Data structures and Algorithm Analysis in C, 2nd edition, M. A. Weiss, Pearson.

3. Lipschutz: Schaum’s outline series Data structures Tata McGraw-Hill

4. Data Structure through C in Depth. S.K. Srivastava and Deepali Srivastava, B.P.B Publication.

Course Name: DISCRETE STRUCTURES

Course Code: BSCHCOSC202

Course Type: **Core (Theoretical)**	Course Details: **CC-4**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.*
2. *Understand the basics of combinatories, and be able to apply the methods from these subjects in problem solving.*
3. *Be able to use effectively algebraic techniques to analyse basic discrete structures and algorithms.*
4. *Understand asymptotic notation, its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples.*
5. *Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.*

Course Content:

Theory

UNIT I. **Sets**: Finite and Infinite Sets, Uncountable Infinite Sets; **Functions:** Domain, Co-domain, Range, Equal function, Exponential function, Logarithmic function, Square function, Cube function, Relations: Reflexive, Symmetric, Anti-symmetric, Properties of Binary

Relations, Closure, Partial Ordering Relations; Counting - Pigeonhole Principle, **Permutation and Combination**: Introduction to Permutation and Combination, Permutation of thing not all different, Multiplication Principle, Addition Principle; **Mathematical Induction**, Principle of Inclusion and Exclusion.

UNIT II. Growth of Functions: Asymptotic Notations, Summation Formulas and Properties, Bounding Summations, Approximation by Integrals.

UNIT III. Recurrences: Recurrence Relations, Generating Functions, Linear Recurrence Relations with Constant Coefficients and their Solution, Substitution Method, Recurrence Trees, Master Theorem.

UNIT IV. Graph Theory: Basic Terminology, Models and Types, Multigraphs and Weighted Graphs, Directed Graph, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and Properties of Trees, Introduction to Spanning Trees.

UNIT V. Propositional Logic: Proposition Or Statements, Truth table, Logical Connectives, Well-formed Formulas, Tautologies, Contradiction, Equivalences, Inference Theory, Conjunctive Normal Form, Disjunctive Normal Form.

References/ Suggested Readings:

1. C.L. Liu & Mahopatra, Elements of Discrete mathematics, 2nd Sub Edition 1985, Tata McGraw Hill.

2. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition, McGraw Hill 2006

3. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms 1988 John wiley Publication.

Course Name: DATA STRUCTURE

Course Code: BSCHCOSGE201

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-2**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles*
2. *To have knowledge of complexity of basic operations like insert, delete, search on these data structures.*
3. *Ability to choose a data structure to suitably model any data used in computer applications.*
4. *Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.*
5. *Ability to assess efficiency tradeoffs among different data structure implementations.*
6. *Implement and know the applications of algorithms for sorting, searching etc.*

Course Content:

Theory

UNIT I. Basic concepts- Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis, Linear and Non Linear data structures, Singly Linked Lists-Operations, Concatenating, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

UNIT II. Stack- Definition and Operations, Array and Linked Implementations, Applications - Valid Expression Checking (Parenthesis matching), Reversal of string, Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation.

UNIT III. Queue - Definition and Operations, Array and Linked Implementations, Applications, Circular Queues - Insertion and Deletion Operations, Dequeue (Double Ended Queue) - Introduction.

UNIT IV. Sorting Methods – Bubble, Insertion, Selection, Shell, Using Divide-Conquer Approach (Quick and Merge sort), Comparison of Sorting Methods, Searching Methods – Linear and Binary.

UNIT V. Trees, Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees, Binary Search tree - Creation, Insertion, Deletion and Search, AVL tree-Definition, Examples, Insertion and Rotations, B tree, B+ tree, Priority Queue- Definition and Implementation, Heap- Definition, Min heap, Max heap, Insertion and Deletion.

UNIT VI. Graphs, Graph ADT, Graph Representations, Graph Traversals and Searching, Static Hashing- Introduction, Hash tables, Hash functions, Overflow Handling.

Practical

Students are required to write and practically execute programs to solve problem using various data structures. The teacher can suitably device problems which help students experiment using the suitable data structures and operations. Some of the problems are indicated below.

1. Write program that uses functions to perform the following:
   a) Creation of list of elements where the size of the list, elements to be inserted and deleted are dynamically given as input.
   b) Implement the operations, insertion, deletion at a given position in the list and search for an element in the list
   c) To display the elements in forward / reverse order
2. Write a program to implement stack data structure and basic operations on it (Insertion, deletion). Write a program that demonstrates the application of stack operations (Eg: infix expression to postfix conversion, postfix evaluation).
3. Write a program to implement queue data structure and basic operations on it (Insertion, deletion, find length) and code at least one application using queues.
4. Write program that implements linear and binary search methods of searching for an elements in a list.
5. Write and trace programs to understand the various phases of sorting elements using the methods a) Bubble sort b) Insertion Sort c) Quicksort etc.
6. Write a program to create a Binary search tree and insert and delete from the tree. Write recursive and non-recursive routines to traverse a binary tree in preorder, inorder and postorder.
7. Write programs for recursion (Eg. Fibonacci numbers, Towers of Hanoi).
8. Represent suitably a graph data structure and demonstrate operations of traversals on it.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. Fundamentals of Data structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson-Freed, Universities Press.
2. Data structures and Algorithm Analysis in C, 2nd edition, M. A. Weiss, Pearson.
3. Lipschutz: Schaum’s outline series Data structures Tata McGraw-Hill
4. Data Structure through C in Depth. S.K. Srivastava and Deepali Srivastava, B.P.B Publication.

Semester- III

Course Name: Operating System

Course Code: BSCHCOSC301

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-5**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Describe the important computer system resources and the role of operating system in their management policies and algorithms.*
2. *To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time.*
3. *Understanding of design issues associated with operating systems.*
4. *Understand various process management concepts including scheduling, synchronization, and deadlocks.*
5. *To have a basic knowledge about multithreading.*
6. *To understand concepts of memory management including virtual memory.*
7. *To understand issues related to file system interface and implementation, disk management.*
8. *To understand and identify potential threats to operating systems and the security features design to guard against them.*
9. *To have sound knowledge of various types of operating systems including Unix and Android.*
10. *Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.*

Course Content:

Theory

UNIT I. (Introduction to Operating System) What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT II. (Operating System Organization and Process Characterization) Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling, Non-Pre-emptive and Preemptive Scheduling Algorithms.

UNIT III. Process Management (Deadlock) Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

UNIT IV. (Inter Process Communication and Synchronization) Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter-process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT V. (Memory Management) Physical and Virtual Address Space; Memory Allocation Strategies– Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory; Page Replacement Algorithms.

UNIT VI. (File and I/O Management, Disk Scheduling, OS security) Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK), Security Policy Mechanism, Protection, Authentication and Internal Access Authorization.

UNIT VII. (Introduction to Android Operating System) Introduction to Android Operating System, Android Development Framework, Android Application Architecture, Android Process Management and File System.

Practical

UNIT I. Students are required to write and practically execute programs to solve following problems using C programming language.

1. WRITE A PROGRAM (using fork() and/or exec() commands) where parent and child execute: a) same program, same code. b) same program, different code. c) before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
5. WRITE A PROGRAM to copy files using system calls.
6. Write programs to implement scheduling algorithms (FCFS, Round Robin, SJF, SRJF)
7. Write program to implement non-preemptive priority based scheduling algorithm.
8. Write program to implement preemptive priority based scheduling algorithm.
9. Write program to calculate sum of n numbers using thread library.
10. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

UNIT II. UNIX and Shell Scripts

1. External and internal commands of UNIX
2. What is shell and various type of shell, Various editors present in unix/linux
3. Different modes of operation in vi editor
4. What is shell script, Writing and executing the shell script
5. Shell variable (user defined and system variables)
6. System calls, Using system calls
7. Pipes and Filters
8. Decision making in Shell Scripts (If else, switch), Loops in shell
9. Functions
10. Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles 2008 5th Edition, Prentice Hall of India. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.
5. Sumitabha, Das, Unix Concepts and Applications, Tata McGraw-Hill Education.
6. Nemeth Synder and Hein, Linux Administration Handbook, Pearson Education, 2nd Edition ,2010.
7. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition, 2014.
8. Yashavant Kanetkar , UNIX Shell Programming, BPB Publication.
9. Kernighan and Pike, The Unix Programming Environment, Prentice-Hall.

Course Name: Analysis of Algorithms

Course Code: BSCHCOSC302

<table>
<tr><td>Course Type:
<b>Core (Theoretical)</b></td><td colspan="3">Course Details: <b>CC-6</b></td><td colspan="2">L-T-P: <b>5 - 1 - 0</b></td></tr>
<tr><td rowspan="3">Credit: <b>6</b></td><td rowspan="3">Full Marks:
<b>50</b></td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>….</td><td><b>10</b></td><td>….</td><td><b>40</b></td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To learn good principles of algorithm design;*
2. *To learn how to analyse algorithms and estimate their worst-case and average case behaviour (in easy cases);*
3. *To become familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles;*

Course Content:

Theory

UNIT I. Introduction: Basic Design and Analysis Techniques of Algorithms, Correctness of Algorithm. Growth of Functions: Asymptotic notation, Big-O, Theta, Omega notations. Algorithm Design Techniques: Iterative Techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

UNIT II. Greedy algorithm: Characteristics and features of problem solving by greedy algorithm, basic structure, feasibility, Huffman code, Dijkstra. Knapsack problem (0/1, fractional)

UNIT III. Sorting and Searching Techniques: Elementary Sorting techniques– Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques- Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques- Medians & Order Statistics, complexity analysis

UNIT IV. Graphs Algorithms: Graph Algorithms– Breadth First Search, Depth First Search and its Applications, Spanning tree, Minimum Spanning Trees (Kruskal and Prism algorithms). String Processing (KMP Technique)

UNIT V. Lower Bounding Techniques: Decision Trees, Balanced Trees(AVL,B tree, Red-Black Trees)

UNIT VI. Advanced Analysis Technique: Randomized Algorithm, Distributed Algorithm, Heuristics

References/ Suggested Readings:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sara basse & A.V. Gelder, Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999
3. Knuth Donald E., The art of computer programming: Fundamental algorithms (Vol. 1), Pearson. 3rd Edition.

Course Name: Computer Networks

Course Code: BSCHCOSC303

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-7**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the structure of Data Communications System and its components. Be familiarize with different network terminologies.*
2. *Familiarize with contemporary issues in network technologies.*
3. *Know the layered model approach explained in OSI and TCP/IP network models*
4. *Identify different types of network devices and their functions within a network.*
5. *Learn basic routing mechanisms, IP addressing scheme and internetworking concepts.*
6. *Familiarize with IP and TCP Internet protocols.*
7. *To understand major concepts involved in design of WAN, LAN and wireless networks.*
8. *Learn basics of network configuration and maintenance.*
9. *Know the fundamentals of network security issues.*

Course Content:

Theory

UNIT I. Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers.

UNIT II. Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation - Multiplexing Techniques-FDM, TDM, Transmission Media.

UNIT III. Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, ConnectionOriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.

UNIT IV. Data Link Layer Functions and Protocol: Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

UNIT V. Multiple Access Protocol and Network Layer: CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back-Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols, Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

UNIT VI. Transport Layer and Application Layer Functions and Protocols: Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Overview of Application Layer Protocol, Overview of DNS Protocol; Overview of WWW & HTTP Protocol.

UNIT VII. Security: Firewall**,** Basics of cryptography; message security; digital signature.

Practical

UNIT I. Network Devices and Configuration

1. Identification of network devices like hub, switch, modem etc.
2. Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities.
3. Configure LAN
4. Configure IP static routing.
5. Configure IP routing using RIP.

UNIT II. All programs should be developed in C/ C++ / Java / Python

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.
3. Simulate and implement stop and wait protocol for noisy channel.
4. Simulate and implement go back N sliding window protocol.
5. Simulate and implement selective repeat sliding window protocol.
6. Simulate and implement MST construction (Prim‟s, Kruskal‟s) for Ethernet
7. Simulate and implement the various routing algorithms (RIP, Distance-Vector routing, Dijkstra‟s, Bellman-Ford, Floyd-Warshall, Flooding)
8. Socket Programming .

Internal (CA) Evaluation: Practical Note Book (15 marks), One experiments from Unit I (10 marks), Viva-voce (5 marks)

ESE Evaluation: One experiment from Unit II (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
2. A. S. Tanenbaum: Computer Networks, Fifth edition, PHI Pvt. Ltd 2011
3. William Stallings:Data and Computer Communications, Eight Edition, Pearson.

Course Name: Operating System and Shell Scripts

Course Code: BSCHCOSGE301

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-3**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Describe the important computer system resources and the role of operating system in their management policies and algorithms.*
2. *To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time.*
3. *Understanding of design issues associated with operating systems.*
4. *Understand various process management concepts including scheduling, synchronization, and deadlocks.*
5. *To have a basic knowledge about multithreading.*
6. *To understand concepts of memory management including virtual memory.*
7. *To understand issues related to file system interface and implementation, disk management.*
8. *To understand and identify potential threats to operating systems.*
9. *To have sound knowledge of various types of operating systems including Unix.*
10. *Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.*

Course Content:

Theory

UNIT I. (Introduction to Operating System) What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems–

Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT II. (Operating System Organization and Process Characterization) Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling, Non-Pre-emptive and Preemptive Scheduling Algorithms.

UNIT III. Process Management (Deadlock) Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

UNIT IV. (Inter Process Communication and Synchronization) Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter-process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT V. (Memory Management) Physical and Virtual Address Space; Memory Allocation Strategies– Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT VI. (File and I/O Management, Disk Scheduling, OS security) Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK), Security Policy Mechanism, Protection, Authentication and Internal Access Authorization

Practical

UNIT I. External and internal commands of UNIX/Linux
UNIT II. What is shell and various type of shell, Various editors present in Unix/Linux
UNIT III. Different modes of operation in vi editor
UNIT IV. What is shell script, Writing and executing the shell script
UNIT V. Shell variable (user defined and system variables)
UNIT VI. System calls, Using system calls
UNIT VII. Pipes and Filters
UNIT VIII. Decision making in Shell Scripts (If else, switch), Loops in shell
UNIT IX. Functions
UNIT X. Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.

2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. W. Stallings, Operating Systems, Internals & Design Principles 2008 5th Edition, Prentice Hall of India. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.
4. Sumitabha, Das, Unix Concepts and Applications, Tata McGraw-Hill Education.
5. Yashavant Kanetkar , UNIX Shell Programming, BPB Publication.
6. Kernighan and Pike, The Unix Programming Environment, Prentice-Hall.

Course Name: Introduction to Computer Networks

Course Code: BSCHCOSGE302

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-3**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the structure of Data Communications System and its components. Be familiarize with different network terminologies.*
2. *Familiarize with contemporary issues in network technologies.*
3. *Know the layered model approach explained in OSI and TCP/IP network models*
4. *Identify different types of network devices and their functions within a network.*
5. *Learn basic routing mechanisms, IP addressing scheme and internetworking concepts.*
6. *Familiarize with IP and TCP Internet protocols.*
7. *To understand major concepts involved in design of WAN, LAN and wireless networks.*
8. *Learn basics of network configuration and maintenance.*

Course Content:

Theory

UNIT I. Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers.

UNIT II. Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation - Multiplexing Techniques- FDM, TDM, Transmission Media.

UNIT III. Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, ConnectionOriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.

UNIT IV. Data Link Layer Functions and Protocol: Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

UNIT V. Multiple Access Protocol and Network Layer: CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back-Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

UNIT VI. Transport Layer and Application Layer Functions and Protocols: Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Overview of Application Layer Protocol, Overview of DNS Protocol; Overview of WWW & HTTP Protocol.

Practical

UNIT I. Network Devices and Configuration

1. Identification of network devices like hub, switch, modem etc.
2. Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities.
3. Configure LAN
4. Configure IP static routing.
5. Configure IP routing using RIP.

UNIT II. All programs should be developed in C/ C++

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.
3. Simulate and implement stop and wait protocol for noisy channel.
4. Simulate and implement go back N sliding window protocol.
5. Simulate and implement selective repeat sliding window protocol.

Internal (CA) Evaluation: Practical Note Book (15 marks), One experiments from Unit I (10 marks), Viva-voce (5 marks)

ESE Evaluation: One experiment from Unit II (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
2. A. S. Tanenbaum: Computer Networks, Fifth edition, PHI Pvt. Ltd 2011.
3. William Stallings:Data and Computer Communications, Eight Edition, Pearson.

Course Name: Programming in Java

Course Code: BSCHCOSSE301

Course Type: **SE (Practical)**	Course Details: **SEC-1**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Knowledge of the structure and model of the Java programming language,*
2. *Use the Java programming language for various programming technologies*
3. *Develop software in the Java programming language,*
4. *Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.*

Course Content:

Practical

UNIT I. Introduction: Java Essentials, Its characteristics, Execution and Compilation, Data types, Variables, Control Statements, Standard Input/ Output.

UNIT II. Constructors, Object Oriented Concepts: Encapsulation, Abstraction, Inheritance, Polymorphisms, JAVA Packages.

UNIT III. Exception Handling, Wrapper Classes, Autoboxing, Multi-thread Programming.

UNIT IV. Applets, Event Handling, AWT.

Students are required to implement object-oriented paradigm using JAVA. Below is the list of some of the experiments.

Part A

1. Program on strings: Check the equality of two strings, Reverse a string.

2. Program using loops: to find the sum of digits of a given number, display a multiplication table, display all prime numbers between 1 to 1000.

3. Program to demonstrate all math class functions.

Part B

4. Program on files: to copy a file to another file using Java to package classes.

5. Program to demonstrate method over-riding and overloading

6. Programs on inheritances.

7. Multi-threaded programming.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (10 marks)

References/ Suggested Readings:

1. E. Balaguruswamy, Programming with Java, 4th Edition, McGraw Hill.2009.
2. John R. Hubbard,"Programming with JAVA, Schaum's Series, 2nd Edition, 2004.
3. Herbert Schildt, The Complete Reference Java 2, TMH.
4. Y. Daniel Liang, Introduction to Java Programming and Data Structures, Pearson, 12th Edition, 2020.

Course Name: Python Programming

Course Code: BSCHCOSSE302

Course Type: **SE (Practical)**	Course Details: **SEC-1**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Develop and Execute simple Python programs.*
2. *Structure a Python program into functions.*
3. *Using Python lists, tuples to represent compound data*
4. *Develop Python Programs for file processing*

Course Content:

Practical

UNIT I. Introduction to Python, Python, Features of Python, Execution of a Python, Program, Writing Our First Python Program, Data types in Python. Python Interpreter and Interactive Mode; Values and Types: int, float, boolean, string, and list; Variables, Expressions, Statements, Tuple Assignment, Precedence of Operators, Comments; Modules and Functions, Function Definition and use, Flow of Execution, Parameters and Arguments

UNIT II. Operators in Python, Input and Output, Control Statements. Boolean Values and operators, Conditional (if), Alternative (if-else), Chained Conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful Functions: Return Values, Parameters, Local and Global Scope, Function Composition, Recursion

UNIT III. Arrays in Python, Strings and Characters. Strings: String Slices, Immutability, String Functions and Methods, String Module; Lists as Arrays. Illustrative Programs: Square Root, gcd, Exponentiation, Sum an Array of Numbers, Linear Search, Binary Search.

UNIT IV. Functions, Lists and Tuples. List Operations, List Slices, List Methods, List Loop, Mutability, Aliasing, Cloning Lists, List Parameters; Tuples: Tuple Assignment, Tuple as Return Value; Dictionaries: Operations and Methods; Advanced List Processing - List Comprehension; Illustrative Programs: Selection Sort, Insertion Sort, Merge sort, Histogram.

UNIT V. Files and Exception: Text Files, Reading and Writing Files, Format Operator; Command Line Arguments, Errors and Exceptions, Handling Exceptions, Modules, Packages; Illustrative Programs: Word Count, Copy File.

The students are required to verify their ability to use core programming basics and program design with functions using Python programming language. The teacher shall programs to strengthen the practical expertise of the students. The following is an indicative list of programs that can be practised.

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format “Sat Oct 11 02:26:23 IST 2020”
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to construct the different pattern, using a nested for loop, Like

```
*
* *
* * *
* *
 *
```

10. Write a Python script that prints prime numbers less than 20.

11. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
12. Write a python program to define a module and import a specific function in that module to another program.
13. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
14. Write a Python class to convert an integer to a roman numeral.
15. Write a Python class to reverse a string word by word.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (10 marks)

References/ Suggested Readings:

1. Alex Martelli, Python in a Nutshell, Oreilly Publication.
2. Allen Downey, Think Python, Green Tea Press.
3. Wesley J. Chun, Core Python Programming, Pearson Education.
4. Mark Lutz, Learning Python, Oreilly Publication.
5. Kenneth A. Lambert, Fundamentals of Python: First Programs, Course Technology Inc.

Semester- IV

Course Name: Software Engineering

Course Code: BSCHCOSC401

Course Type: **Core (Theoretical)**	Course Details: **CC-8**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Basic knowledge and understanding of the analysis and design of complex systems.*
2. *Ability to apply software engineering principles and techniques.*
3. *To produce efficient, reliable, robust and cost-effective software solutions.*
4. *Ability to work as an effective member or leader of software engineering teams.*
5. *To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.*

Course Content:

Theory

UNIT I. Software Development Approaches: Introduction; Evolving Role of Software; Software Characteristics; Software Applications. Software Design Processes: Introduction; What is Meant by Software Engineering? Definitions of Software Engineering; The Serial or Linear Sequential Development Model; Iterative Development Model; The incremental Development Model

UNIT II. Software Design Principles: Introduction, System Models: Data-flow Models, Semantic Data Models, Object Models, Inheritance Models, Object Aggregation, Service Usage Models, Data Dictionaries; Software Design: The Design Process, Design Methods, Design description, Design Strategies, Design Quality; Architectural Design: System Structuring, The Repository Model, The Client–Server Model, The Abstract Machine Model, Control Models, Modular Decomposition, Domain-Specific Architectures.

UNIT III. Object Oriented Design: Introduction; Object Oriented Design: Objects, Object Classes & Inheritance, Inheritance, Object Identification, An Object -Oriented Design Example, Object Aggregation; Service Usage; Object Interface Design: Design Evolution, Function Oriented Design, Data–Flow Design; Structural Decomposition: Detailed Design.

UNIT IV. An Assessment of Process Life-Cycle Models: Introduction; Overview of the Assessment of Process; The Dimension of Time; The Need for a Business Model in Software Engineering; Classic Invalid Assumptions: First Assumption: Internal or External Drivers, Second Assumption: Software or Business Processes, Third Assumption: Processes or Projects, Fourth Assumption: Process Centered or Architecture Centered; Implications of the New Business Model; Role of the Problem - Solving Process in this Approach: Data, Problem Definition, Tools and Capabilities; Redefining the Software Engineering Process: Round-Trip Problem-Solving Approach, Activities, Goals, Interdisciplinary Resources, Time.

UNIT V. Software Reliability: Introduction; Software Reliability Metrics; Programming for Reliability: Fault Avoidance, Fault Tolerance, Software Reuse.

UNIT VI. Software Testing Techniques: Introduction; Software Testing Fundamental; Testing Principles; White Box Testing; Control Structure Testing; Black Box Testing; Boundary Value Analysis; Testing GUIs; Testing Documentation and Help Facilities; Software Testing Strategies: Introduction; Organizing for Software Testing; Software Testing Strategy, Unit Testing: Unit Test Considerations, Top-Down Integration, Bottom-Up Integration.

References/ Suggested Readings:

1. R. G. Pressman, Software Engineering, TMH.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Learning Pvt. Ltd.
3. Sommerville, Ian, Software Engineering, Pearson Education
4. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publications.
5. Pfleeger, Shari Lawrence, Software Engineering Theory and Practice, Second edition. Prentice- Hall 2001.

Course Name: Database Management System

Course Code: BSCHCOSC402

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-9**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Gain knowledge of database systems and database management systems software*
2. *Ability to model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model.*
3. *Formulate, using SQL, solutions to a broad range of query and data update problems.*
4. *Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.*

5. *Be acquainted with the basics of transaction processing and concurrency control.*
6. *Familiarity with database storage structures and access techniques.*
7. *Compare, contrast and analyse the various emerging technologies for database systems.*
8. *Analyse strengths and weaknesses of the applications of database technologies to various subject areas*

Course Content:

Theory

UNIT I. Basic Database Concepts, Terminology, and Architecture; Types of Database Management Systems. Differences between Relational and other Database Models. Data Modelling: Relations, Schemas, Constraints, Queries, and Updates; Conceptual vs. Physical Modeling; Entity Types, attributes, ER Diagrams.

UNIT II. SQL Data Definition: Specifying Tables, Data Types, Constraints; Simple SELECT, INSERT, UPDATE, DELETE Statements; Complex SELECT Queries, including Joins and Nested Queries; Actions and Triggers; Views; Altering Schemas.

UNIT III. Relational Algebra: Definition of Algebra; Relations as Sets; Operations: SELECT, PROJECT, JOIN, etc. Normalization Theory and Functional Dependencies, 2NF, 3NF, BCNF, 4NF, 5NF;

UNIT IV. Indexing: Files, Blocks, and Records, Hashing; RAID; Replication; Single-Level and Multi-Level Indexes; B-Trees and B+-Trees. Basics of Transactions, Concurrency and Recovery.

UNIT V. DATABASE PROGRAMMING: Embedded SQL; Dynamic SQL, Avoiding Injection Attacks; Stored Procedures;

UNIT VI. BIG DATA: Motivations; OLAP vs. OLTP; Batch Processing; Map Reduce and Hadoop; Spark;

Practical

Students are required to practice the concepts learnt in the theory by designing and querying a database for a chosen organization (Like Library, Transport etc). The teacher may devise appropriate weekly lab assignments to help students practice the designing, querying a database in the context of example database. Some indicative list of experiments is given below.

Experiment 1: E-R Model Analyze the organization and identify the entities , attributes and relationships in it. . Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Experiment 2: Concept design with E-R Model Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any).

Experiment 3: Relational Model Represent all the entities (Strong, Weak) in tabular fashion. Represent relation ships in a tabular fashion.

Experiment 4: Normalization Apply the First, Second and Third Normalization levels on the database designed for the organization

Experiment 5: Installation of Mysql/Postgresql/SQL Server/Oracle

Experiment 6: Practicing DDL commands. Creating databases, How to create tables, altering the database, dropping tables and databases. Try truncate, rename commands etc.

Experiment 7: Practicing DML commands on the Database created for the example organization DML commands are used to for managing data within schema objects. Some examples: ● SELECT - retrieve data from the a database ● INSERT - insert data into a table ● UPDATE - updates existing data within a table ● DELETE - deletes all records or few records from a table.

Experiment 8: Querying practice queries (along with sub queries) involving ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Experiment 9: Practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Experiment 10: Triggers - Work on Triggers. Creation of, insert trigger, delete trigger, update trigger.

Internal (CA) Evaluation: Practical Note Book (15 marks), Experiment (10 marks) – Experiment 1 to Experiment 5, Viva-voce (5 marks)

ESE Evaluation: Experiment (10 marks) – Experiment 6 to Experiment10, Viva-voce (10 marks)

References/ Suggested Readings:

1. Elmasri's and Navathe's Fundamentals of Database Systems. Addison-Wesley
2. Raghu Ramakrishnan, Johannes Gehrke, Data base Management Systems, McGraw Hill Education
3. Silberschatz, Henry. F. Korth, S. Sudarshan, Data base System Concepts, McGraw Hill Education
4. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.

Course Name: Object Oriented Programming

Course Code: BSCHCOSC403

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-10**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn the concepts of data, abstraction and encapsulation*
2. *Be able to write programs using classes and object.*
3. *Understand conceptually principles of Inheritance and Polymorphism and their use and program level implementation.*
4. *Learn exception and basic event handling mechanisms in a program.*
5. *Learn generic programming with class templates and function templates.*
6. *To learn typical object-oriented constructs of specific object oriented programming language*

Course Content:

Theory

UNIT I. Basics: Introduction to Object Oriented Programming and its Basic Features, Basic Components of C++, Characteristics of Object-Oriented Language, Structure of a C++ Program, Flow Control Statements in C++, Functions - Scope of Variables, Inline Functions, Recursive Functions, Pointers to Functions, C++ Pointers, Arrays, Dynamic Memory Allocation and De-Allocation

UNIT II. Differences Between Object Oriented and Procedure Oriented Programming, Abstraction, Overview of Object-Oriented Programming Principles, Encapsulation, C++ Classes, Objects, User Defined Types, Constructors and Destructors, this Pointer, Friend Functions, Data Abstraction, Operator Overloading, Type Conversion

UNIT III. Class Inheritance, Base and Derived Classes, Virtual Base Class, Virtual Functions, Polymorphism, Static and Dynamic Bindings, Base and Derived Class Virtual Functions, Dynamic Binding through Virtual Functions, Pure Virtual Functions, Abstract Classes, Virtual Destructors

UNIT IV. Stream Classes Hierarchy, Stream I/O, File Streams, Overloading the Extraction and Insertion Operators, Error Handling during File Operations, Formatted I/O.

UNIT V. Exception Handling- Benefits of Exception Handling, Throwing an Exception, the Try Block, Catching an Exception, Exception Objects, Exception Specifications, Rethrowing an Exception, Uncaught Exceptions.

UNIT VI. Templates - Class Templates and Function Templates, simple generic classes and generic function, simple example programs. Introduction to Standard Template Library (STL), Components of STL, STL-List, Vector, Array.

Practical

Students are required to understand the object-oriented concepts using C++. They are required to practice the concepts learnt in the theory. Some of the programs to be implemented are listed as follows:

1. Number of vowels and number of characters in a string.
2. Write a function called zeros maller() that is passed with two introduce arguments by reference and set the smaller of the number to zero. Write a main() program to access this function.
3. Demonstration of Class, Constructors, destructors, input and output functions, Objects
4. Demonstration of array of object.
5. Demonstration of friend functions.
6. Demonstration of operator overloading.
7. Demonstration of inheritance.
8. Using this pointer to return a value (return by reference).
9. Demonstration of virtual function.
10. Demonstration of static function.
11. Accessing a particular record in a student's file.
12. Demonstration of exception handling.
13. Demonstration of class template and function template
14. Demonstration of Standard Template Library (STL).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks).

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. E.Balagurusamy , Object Oriented Programming through C++,TMH.
2. Lafore Robert, Object Oriented Programming in Turbo C++, Galgotia Publications.
3. Herbert Schildt, C++: The Complete Reference, 4th Edition
4. B. Stroutstrup, The C++ Programming Language, 3rd Edition, Pearson Education.
5. Ashok N Kamthane, Programming in C++, Pearson 2nd Edition.

Course Name: Object Oriented Programming in C++

Course Code: BSCHCOSGE401

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-4**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn the concepts of data, abstraction and encapsulation*
2. *Be able to write programs using classes and objects.*
3. *Understand conceptually principles of Inheritance and Polymorphism and their use and program level implementation.*
4. *Learn exception and basic event handling mechanisms in a program*
5. *Learn generic programming with class templates and function templates.*
6. *To learn typical object-oriented constructs of specific object oriented programming language.*

Course Content:

Theory

UNIT I. Basics: Introduction to Object Oriented Programming and its Basic Features, Basic Components of C++, Characteristics of Object-Oriented Language, Structure of a C++ Program, Flow Control Statements in C++, Functions - Scope of Variables, Inline Functions, Recursive Functions, Pointers to Functions, C++ Pointers, Arrays, Dynamic Memory Allocation and De-Allocation

UNIT II. Differences Between Object Oriented and Procedure Oriented Programming, Abstraction, Overview of Object-Oriented Programming Principles, Encapsulation, C++ Classes, Objects, User Defined Types, Constructors and Destructors, this Pointer, Friend Functions, Data Abstraction, Operator Overloading, Type Conversion

UNIT III. Class Inheritance, Base and Derived Classes, Virtual Base Class, Virtual Functions, Polymorphism, Static and Dynamic Bindings, Base and Derived Class Virtual Functions, Dynamic Binding through Virtual Functions, Pure Virtual Functions, Abstract Classes, Virtual Destructors

UNIT IV. Stream Classes Hierarchy, Stream I/O, File Streams, Overloading the Extraction and Insertion Operators, Error Handling during File Operations, Formatted I/O.

UNIT V. Exception Handling- Benefits of Exception Handling, Throwing an Exception, the Try Block, Catching an Exception, Exception Objects, Exception Specifications.

UNIT VI. Templates - Class Templates and Function Templates, simple generic classes and generic function, simple example programs. Introduction to Standard Template Library (STL).

Practical

Students are required to understand the object-oriented concepts using C++. They are required to practice the concepts learnt in the theory. Some of the programs to be implemented are listed as follows:

1. Number of vowels and number of characters in a string.
2. Write a function called zeros maller () that is passed with two introduce arguments by reference and set the smaller of the number to zero. Write a main() program to access this function.
3. Demonstration of Class, Constructors, destructors, input and output functions, Objects
4. Demonstration of array of object.
5. Demonstration of friend functions.
6. Demonstration of operator overloading.
7. Demonstration of inheritance.
8. Using this pointer to return a value (return by reference).
9. Demonstration of virtual function.
10. Demonstration of static function.
11. Accessing a particular record in a student's file.
12. Demonstration of exception handling.
13. Demonstration of class template and function template

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. E.Balagurusamy , Object Oriented Programming through C++,TMH.
2. Lafore Robert, Object Oriented Programming in Turbo C++, Galgotia Publications.
3. Herbert Schildt, C++: The Complete Reference, 4th Edition.
4. B. Stroutstrup, The C++ Programming Language, 3rd Edition, Pearson Education.

Course Name: Introduction to Database Management System

Course Code: BSCHCOSGE402

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-4**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Gain knowledge of database systems and database management systems software*
2. *Ability to model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model.*
3. *Formulate, using SQL, solutions to a broad range of query and data update problems.*
4. *Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.*
5. *Be acquainted with the basics of transaction processing and concurrency control.*
6. *Familiarity with database storage structures and access techniques.*
7. *Compare, contrast and analyse the various emerging technologies for database systems.*
8. *Analyse strengths and weaknesses of the applications of database technologies to various subject areas*

Course Content:

Theory

UNIT I. Basic Database Concepts, Terminology, and Architecture; Types of Database Management Systems. Differences between Relational and other Database Models. Data Modelling: Relations, Schemas, Constraints, Queries, and Updates; Conceptual vs. Physical Modeling; Entity Types, attributes, ER Diagrams.

UNIT II. SQL Data Definition: Specifying Tables, Data Types, Constraints; Simple SELECT, INSERT, UPDATE, DELETE Statements; Complex SELECT Queries, including Joins and Nested Queries; Actions and Triggers; Views; Altering Schemas.

UNIT III. Relational Algebra: Definition of Algebra; Relations as Sets; Operations: SELECT, PROJECT, JOIN, etc. Normalization Theory and Functional Dependencies, 2NF, 3NF, BCNF, 4NF, 5NF;

UNIT IV. Indexing: Files, Blocks, and Records, Hashing; RAID; Replication; Single-Level and Multi-Level Indexes; B-Trees and B+-Trees. Basics of Transactions, Concurrency and Recovery.

UNIT V. Introduction to BIG DATA: Motivations; Applications.

Practical

Students are required to practice the concepts learnt in the theory by designing and querying a database for a chosen organization (Like Library, Transport etc). The teacher may devise appropriate weekly lab assignments to help students practice the designing , querying a database in the context of example database. Some indicative list of experiments is given below.

Experiment 1: E-R Model Analyze the organization and identify the entities , attributes and relationships in it. . Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Experiment 2: Concept design with E-R Model Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any).

Experiment 3: Relational Model Represent all the entities (Strong, Weak) in tabular fashion. Represent relation ships in a tabular fashion.

Experiment 4: Normalization Apply the First, Second and Third Normalization levels on the database designed for the organization

Experiment 5: Installation of Mysql/Postgresql/SQL Server/Oracle

Experiment 6: Practicing DDL commands. Creating databases, How to create tables, altering the database, dropping tables and databases. Try truncate, rename commands etc.

Experiment 7: Practicing DML commands on the Database created for the example organization DML commands are used to for managing data within schema objects. Some examples: ● SELECT - retrieve data from the a database ● INSERT - insert data into a table ● UPDATE - updates existing data within a table ● DELETE - deletes all records or few records from a table.

Experiment 8: Querying practice queries (along with sub queries) involving ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Experiment 9: Practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Experiment 10: Triggers - Work on Triggers. Creation of, insert trigger, delete trigger, update trigger.

Internal (CA) Evaluation: Practical Note Book (15 marks), Experiment (10 marks) – Experiment 1 to Experiment 5, Viva-voce (5 marks)

ESE Evaluation: Experiment (10 marks) – Experiment 6 to Experiment10, Viva-voce (10 marks)

References/ Suggested Readings:

1. Elmasri's and Navathe's Fundamentals of Database Systems. Addison-Wesley.
2. Raghu Ramakrishnan, Johannes Gehrke, Data base Management Systems, McGraw Hill Education.
3. Silberschatz, Henry. F. Korth, S. Sudarshan, Data base System Concepts, McGraw Hill Education.
4. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.

Course Name: Mobile Application Development

Course Code: BSCHCOSSE401

Course Type: **SE (Practical)**	Course Details: **SEC-2**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To understand Android platform and its architecture.*
2. *To learn about mobile devices types and different modern mobile operating systems.*
3. *To learn activity creation and Android User Interface designing.*
4. *To learn basics of Intent, Broadcast and Internet services.*
5. *To learn about different wireless mobile data transmission standards.*
6. *To understand and learn how to integrate basic phone features, multimedia, camera and Location based services in Android Application.*
7. *To learn about different systems for mobile application development, deployment and distribution in Mobile market place (Android, IoS).*
8. *To understand and carry out functional test strategies for mobile applications.*

Course Content:

Practical

Students are directed to do a minor project based on the contents of the course below (UNIT I to UNIT VI) for internal and ESE evaluation.

UNIT I. (Introduction) What is Android, Android Versions and its Feature Set, Various Android Devices on the Market, Android Market Application Store, Android Development Environment System Requirements, Android SDK, Installing Java, and ADT bundle - Eclipse Integrated Development Environment (IDE), Creating Android Virtual Devices (AVDs)

UNIT II. (Android Architecture Overview and Application) Android Software Stack, The Linux Kernel, Android Runtime - Dalvik Virtual Machine, Android Runtime – Core Libraries, Dalvik VM Specific Libraries, Java Interoperability Libraries, Android Libraries, Application Framework, Creating a New Android Project ,Defining the Project Name and SDK Settings, Project Configuration Settings, Configuring the Launcher Icon, Creating an Activity, Running the Application in the AVD, Stopping a Running Application, Modifying the Example Application, Reviewing the Layout and Resource Files,

UNIT III. (Android Software Development Platform and Framework) Understanding Java SE and the Dalvik Virtual Machine, The Directory Structure of an Android Project, Common Default Resources Folders, The Values Folder, Leveraging Android XML, Screen Sizes , Launching Mobile Application: The AndroidManifest.xml File, Android Application Components, Android Activities: Defining the UI, Android Service s: Processing in the Background, Broadcast Receivers: Announcements and Notifications Content Providers: Data Management, Android Intent Objects: Messaging for Components, Android Manifest XML: Declaring Your Components

UNIT IV. (Understanding Android User Interfaces, Views and Layouts) Designing for Different Android Devices, Views and View Groups, Android Layout Managers, The View Hierarchy, Designing an Android User Interface using the Graphical Layout Tool Displaying Text with TextView, Retrieving Data from Users, Using Buttons, Check Boxes and Radio Groups, Getting Dates and Times from Users, Using Indicators to Display Data to Users, Adjusting Progress with Seek Bar, Working with Menus using views, Gallery, Image Switcher, Grid View, and Image View views to display images, Creating Animation

UNIT V. (Databases, Intents, Location-based Services) Saving and Loading Files, SQLite Databases, Android Database Design, Exposing Access to a Data Source through a Content Provider, Content Provider Registration, Native Content Providers Intents and Intent Filters: Intent Overview, Implicit Intents, Creating the Implicit Intent Example Project, Explicit Intents, Creating the Explicit Intent Example Application, Intents with Activities, Intents with Broadcast Receivers

UNIT VI. Sending SMS Messages Programmatically, Getting Feedback after Sending the Message Sending SMS Messages Using Intent Receiving, sending email, Introduction to location-based service, configuring the Android Emulator for Location -Based Services, Geocoding and Map-Based Activities Multimedia: Audio, Video, Camera: Playing Audio and Video, Recording Audio and Video, Using the Camera to Take and Process Pictures.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks)

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Harwani, Android Programming Unleashed (1 st Edition), SAMS.
2. Richard Rodger, Beginning Mobile Application Development in the Cloud (2011), Wrox.

Course Name: Web Programming

Course Code: BSCHCOSSE402

Course Type: **SE (Practical)**	Course Details: **SEC-2**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To understand basics of the Internet and World Wide Web*
2. *To acquire knowledge and skills for creation of web site considering both client and server-side programming*
3. *To learn basic skill to develop responsive web applications*
4. *To understand different web extensions and web services standards*
5. *To understand basic concepts of Search Engine Basics.*
6. *To learn Web Service Essentials.*
7. *To learn Rich Internet Application Technologies.*
8. *To understand and get acquainted with Web Analytics 2.0*
9.

Course Content:

Practical

Students are directed to do a minor project based on the contents of the course below (UNIT I to UNIT VI) for internal and ESE evaluation.

UNIT I. (Introduction to World Wide Web) -Internet Standards, Introduction to WWW and WWW Architecture, Internet Protocols, Overview of HTTP, HTTP request – response, Generations of dynamic web pages.

UNIT II. (User Interface Design) Introduction to HTML and HTML5, TML Tags, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms. The need for CSS, Introduction to CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style, Backgrounds, Manipulating Text, Margins and Padding, Positioning using CSS.

UNIT III. (Java Programming) Java Script, Introduction, Core features, Data types and Variables, Operators, Expressions, Functions, Objects, Array, Date and Math related Objects. JAVA Networking classes, TCP/IP Protocol Suite, File Transfer Protocol (FTP), Java Environment |Setup for Web Applications, JavaBean, Application Builder Tool, Bean Developer Kit (BDK), The Java Beans API, Introduction to EJB

UNIT IV. (Database) Database basics, SQL, MySQL, PostgreSQL, JDBC API, Driver Types, Two-tier and Three-tier Models, Connection Overview, Transactions, Driver Manager Overview, Statement Overview, Result Set Overview, Types of Result Sets, Concurrency Types, Prepared Statement Overview

UNIT V. (Java Applet and JSP) Java Web Programs and Applets, Web Application, Servlet, Servlet Life Cycle, Servlet Programming, Introduction to JSP, Life Cycle of a JSP Page, Translation and Compilation, Creating Static Content, Response and Page Encoding, Creating Dynamic Content, Using Objects within JSP Pages, JSP Programming

UNIT VI. (Dot Net Framework) Introduction to Dot Net, Dot Net framework and its architecture, CLR, Assembly, Components of Assembly, DLL hell and Assembly Versioning, Overview to C#, Introduction to ASP.net, Asp.net Programming.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks).

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. James Keogh, J2EE: The complete Reference.
2. John Brock, Arun Gupta, Geertjan Wielenga, Java EE and HTML5 Enterprise Application Development (Oracle Press)
3. James Holmes, Struts: The Complete Reference, 2nd Edition
4. Stephen Walther, Kevin Scott Hoffman, Nate Dudek, ASP.NET Unleashed
5. John Sharp, Microsoft Visual C# 2013 Step by Step.
6. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.