CBCS System

Scheme of Examination

&

Syllabus

For

Master of Science (M.Sc.)

Computer Science

Part I & II – Semester I, II, III & IV

SESSION 2022-23

CHRISTIAN EMINENT COLLEGE, INDORE
ACADEMY OF MANAGEMENT, PROFESSIONAL EDUCATION & RESEARCH
An Autonomous Institution Established in 1996
AFFILIATED TO DEVI AHILYA VISHWAVIDYALAYA, INDORE
F-SECTOR, R.S.S. NAGAR, H.I.G. MAIN ROAD, INDORE

Scheme of Examination

M.Sc. (Computer Science) Part I – Semester I

COURSE	CREDITS	TOTAL HOURS	LECTURE HOURS PER WEEK	MIN. GRADE POINT OUT OF 10
CORE COURSE				
MCS-T101 COMPUTER & COMMUNICATION FUNDAMENTALS	03	48	03	04
MCS -T102 PROGRAMMING & PROBLEM SOLVING USING C	04	64	04	04
MCS -T103 OPERATING SYSTEMS	04	64	04	04
MCS -T104 DISCRETE STRUCTURES	04	64	04	04
MCS-T105 COMMUNICATION SKILLS	02	32	02	04
MCS -P106 PRACTICAL COURSE IN OPERATING SYSTEMS	02	32	04	04
MCS -P107 PRACTICAL COURSE USING C	02	32	04	04
SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC)				
MCS – 108 SKEG (ANY ONE)	SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC/GC)			
SKEG-108 HEALTH EDUCATION	03	48	03	04
SKEG-116 MANAGERIAL SKILLS				
SKEG-T119 PERSONALITY DEVELOPMENT				
SKEG-T/P122 PC SOFTWARE & INSTALLATIONS	02+01	32+16	02+02	04
TOTAL	24	384	28/29	

Scheme of Examination

M.Sc. (Computer Science) Part I – Semester I

Course	Max. Marks				Min. Marks		
	External Theory Examination	Internal Theory Examination	Practical Examination	TOTAL MARKS	External Theory Exam.	Internal Theory Exam.	Practical Marks
MCS-T101 COMPUTER & COMMUNICATION FUNDAMENTALS	70	30	-	100	28	12	-
MCS -T102 PROGRAMMING & PROBLEM SOLVING USING C	70	30	-	100	28	12	-
MCS -T103 OPERATING SYSTEMS	70	30	-	100	28	12	-
MCS -T104 DISCRETE STRUCTURES	70	30	-	100	28	12	-
MCS-T105 COMMUNICATION SKILLS	70	30	-	100	28	12	-
MCS -P106 PRACTICAL COURSE IN OPERATING SYSTEMS	-	-	50	50	-	-	20
MCS -P107 PRACTICAL COURSE USING C	-	-	50	50	-	-	20
MCS – T 108 SKEG-T (ANY ONE) SKILL ENHANCEMENT COURSE	70	30	-	100	28	12	-
TOTAL MARKS	420	180	100	700	-	-	
MCS – T/P 108 SKEG-T/P (A.O.) SKILL ENHANCEMENT COURSE	50	30	20	100	20	12	08
TOTAL MARKS	400	180	120	700	-	-	
GRAND TOTAL	700				315		

Scheme of Examination

M.Sc. (Computer Science) Part I – Semester II

COURSE	CREDITS	TOTAL HOURS	LECTURE HOURS PER WEEK	MIN. GRADE POINT OUT OF 10
CORE COURSE				
MCS-T201 DATA STRUCTURES USING C++	04	64	04	04
MCS -T202 DATABASE MANAGEMENT SYSTEM	03	48	03	04
MCS -T203 SOFTWARE ENGINEERING	03	48	03	04
MCS -T204 COMPUTER ARCHITECTURE AND ORGANIZATION	04	64	04	04
MCS -T205 COMPUTER ORIENTED NUMERICAL AND STATISTICAL METHOD	03	48	03	04
MCS-P206 PRACTICAL ON DATA STRUCTURE USING C++ &	02	32	04	04
MCS-P207 PRACTICAL ON DATABASE MANAGEMENT SYSTEMS	02	32	04	04
SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC)				
MCS – 208 SKEG (ANY ONE)	SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC/GC)			
SKEG-T103 COMMUNICATIVE ENGLISH	03	48	03	04
SKEG-T107 FUNDAMENTAL OF BANKING & INSURANCE				
SKEG-T108 HEALTH EDUCATION				
SKEG-T119 PERSONALITY DEVELOPMENT				
TOTAL	24	384	28	

Scheme of Examination

M.Sc. (Computer Science) Part I – Semester II

Course	Max. Marks				Min. Marks		
	External Theory Examination	Internal Theory Examination	Practical Examination	TOTAL MARKS	External Theory Exam.	Internal Theory Exam.	Practical Marks
MCS-T201 DATA STRUCTURES USING C++	70	30	-	100	28	12	-
MCS -T202 DATABASE MANAGEMENT SYSTEM	70	30	-	100	28	12	-
MCS -T203 SOFTWARE ENGINEERING	70	30	-	100	28	12	-
MCS -T204 COMPUTER ARCHITECTURE AND ORGANIZATION	70	30	-	100	28	12	-
MCS -T205 COMPUTER ORIENTED NUMERICAL AND STATISTICAL METHOD	70	30	-	100	28	12	-
MCS-P206 PRACTICAL ON DATA STRUCTURE USING C++ &	-	-	50	50	-	-	20
MCS-P207 PRACTICAL ON DATABASE MANAGEMENT SYSTEMS	-	-	50	50	-	-	20
MCS – T 208 SKEG (ANY ONE) SKILL ENHANCEMENT COURSE	70	30	-	100	28	12	-
TOTAL MARKS	420	180	100	700	-	-	
GRAND TOTAL	700				315		

Scheme of Examination

M.Sc. (Computer Science) Part II – Semester III

COURSE	CREDITS	TOTAL HOURS	LECTURE HOURS PER WEEK	MIN. GRADE POINT OUT OF 10
CORE COURSE				
MCS-T301 OBJECT ORIENTED PROGRAMMING (USING JAVA)	03	48	03	04
MCS –T302 DATABASE APPLICATIONS AND TOOLS	03	48	03	04
MCS –T303 THEORY OF COMPUTATION	03	48	03	04
MCS –T304 COMPUTER GRAPHICS & MULTIMEDIA	03	48	03	04
MCS-T305 COMPUTER NETWORKS	03	48	03	04
MCS –P306 PRACTICAL COURSE IN DATABASE APPLICATION	02	32	04	04
MCS –P307 PRACTICAL COURSE OBJECT ORIENTED PROGRAMMING (USING JAVA)	02	32	04	04
MCS –P308 PRACTICAL COURSE COMPUTER GRAPHICS	02	32	04	04
SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC)				
MCS – 309 SKEG (ANY ONE)	SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC/GC)			
SKEG-T108 HEALTH EDUCATION	03	48	03	04
SKEG-T116 MANAGERIAL SKILLS				
SKEG-T119 PERSONALITY DEVELOPMENT				
SKEG-T/P124 PHP PROGRAMMING	02+01	32+16	02+02	04
TOTAL	24	384	30/31	

Scheme of Examination

Scheme of Examination

M.Sc. (Computer Science) Part II – Semester III

Course	Max. Marks				Min. Marks		
	External Theory Examination	Internal Theory Examination	Practical Examination	TOTAL MARKS	External Theory Exam.	Internal Theory Exam.	Practical Marks
MCS-T301 OBJECT ORIENTED PROGRAMMING (USING JAVA)	70	30	-	100	28	12	-
MCS –T302 DATABASE APPLICATIONS AND TOOLS	70	30	-	100	28	12	-
MCS –T303 THEORY OF COMPUTATION	70	30	-	100	28	12	-
MCS –T304 COMPUTER GRAPHICS & MULTIMEDIA	70	30	-	100	28	12	-
MCS-T305 COMPUTER NETWORKS	70	30	-	100	28	12	-
MCS –P306 PRACTICAL COURSE IN DATABASE APPLICATION	-	-	50	50	-	-	20
MCS –P307 PRACTICAL COURSE OBJECT ORIENTED PROGRAMMING (USING JAVA)	-	-	50	50	-	-	20
MCS –P308 PRACTICAL COURSE COMPUTER GRAPHICS	-	-	50	50	-	-	20
MCS – T 309 SKEGT (ANY ONE) SKILL ENHANCEMENT COURSE	70	30	-	100	28	12	-
TOTAL MARKS	420	180	150	750	-	-	
MCS – T/P 309 SKEG-T/P (A.O.) SKILL ENHANCEMENT COURSE	50	30	20	100	20	12	08
TOTAL MARKS	400	180	170	750	-	-	
GRAND TOTAL	750				338		

Scheme of Examination

M.Sc. (Computer Science) Part II – Semester IV

COURSE	CREDITS	TOTAL HOURS	LECTURE HOURS PER WEEK	MIN. GRADE POINT OUT OF 10
CORE COURSE				
MCS-T401 UNIX /LINUX ADMINISTRATION	03	48	03	04
MCS –T402 COMPILER DESIGN	03	48	03	04
MCS –T403 INTERNET AND WEB TECHNOLOGY	04	64	04	04
MCS –T404 DESIGN AND ANALYSIS OF ALGORITHMS	03	48	03	04
MCS-P405 PRACTICAL ON UNIX /LINUX ADMINISTRATION	02	32	04	04
MCS-P406 PRACTICAL ON INTERNET AND WEB TECHNOLOGY	02	32	04	04
MCS-P407 PRACTICAL ON DESIGN AND ANALYSIS OF ALGORITHMS	02	32	04	04
SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC)				
MCS – 408 SKEG (ANY ONE)	SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC/GC)			
SKEG-T107 FUNDAMENTAL OF BANKING & INSURANCE	03	48	03	04
SKEG-T108 HEALTH EDUCATION				
SKEG-T119 PERSONALITY DEVELOPMENT				
SKEG-T/P117 • NET PROGRAMMING	02+01	32+16	02+02	04
PROJECT / INTERNSHIP				
MCS-P409 PROJECT / INTERNSHIP	02	32	-	04
TOTAL	24	384	28/29	

Scheme of Examination

Scheme of Examination

M.Sc. (Computer Science) Part II – Semester IV

Course	Max. Marks				Min. Marks		
	External Theory Examination	Internal Theory Examination	Practical Examination	TOTAL MARKS	External Theory Exam.	Internal Theory Exam.	Practical Marks
MCS-T401 UNIX /LINUX ADMINISTRATION	70	30	-	100	28	12	-
MCS –T402 COMPILER DESIGN	70	30	-	100	28	12	-
MCS –T403 INTERNET AND WEB TECHNOLOGY	70	30	-	100	28	12	-
MCS –T404 DESIGN AND ANALYSIS OF ALGORITHMS	70	30	-	100	28	12	-
MCS-P405 PRACTICAL ON UNIX /LINUX ADMINISTRATION	-	-	50	50	-	-	20
MCS-P406 PRACTICAL ON INTERNET AND WEB TECHNOLOGY	-	-	50	50	-	-	20
MCS-P407 PRACTICAL ON DESIGN AND ANALYSIS OF ALGORITHMS	-	-	50	50	-	-	20
MCS – T 408 SKEG (ANY ONE) SKILL ENHANCEMENT COURSE	70	30	-	100	28	12	-
MCS-P409 PROJECT / INTERNSHIP	-	-	100	100	-	-	40
TOTAL MARKS	350	150	250	750	-	-	
MCS – T/P 409 SKEG-T/P (A.O.) SKILL ENHANCEMENT COURSE	50	30	20	100	20	12	08
TOTAL MARKS	330	150	270	750	-	-	
GRAND TOTAL	750				338		

Scheme of Examination

Under CBCS System

Part I & II – Semester I, II, III & IV

- Under CBCS System every PG Course and B.Ed. has been distributed in two parts namely – Core and Elective. The subjects related to course are Core and are compulsory. In each semester the students have to opt one Elective Course from prescribed electives.
- The minimum credits for each course are 20 and maximum may be 24. The credits are finalized with the requirements of respective course.
- The total minimum credits for completing the Post Graduate course and B.Ed are 80.
- For each course there will be 70% marks for External Examinations and 30% for Internal Examinations (CCE). The students have to clear both External and Internal Examinations separately.
- The pass marks in individual paper will be **40%** and in aggregate **45%**.
- The subject wise marks obtained by the student will be converted into prescribed 10 Point Grade Scale. The prescribed Grade Scale and related information are available in Examination Rules and for details follow or refer prescribed CBCS Guidelines.
- The students who are **awarded ATKT in two subjects** will be eligible to appear in the examination of next semester. However the student **will not be allowed** to appear in the next semester examination with more than **four ATKT at a time**.
- In case of more than two ATKT in a particular semester will be considered as fail in that semester and the student has to reappear in that particular semester examination.
- ATKT students have to follow the old syllabus but repeaters have to take the examination with the new syllabus.
- A student will have to compulsorily clear a program within **Three Academic Years** including the academic year of the admission, failing which he /she will not be allowed to continue the course. If a student doesn't clear all the semesters of the course in the above three years completely, then all his/ her previous result will be treated as null and void.
- Only those students who clear the program in one attempt and without gap will be eligible for position in the **Merit List**.
- A student who fails in aggregate is permitted to appear in **any one or two** papers of his/her choice to make up for the shortfall in the aggregate. Such a student can also appear in all the papers of that semester as an ex-student, provided the student applies for the same in the beginning of the semester.
- The students who are declared fail in aggregate will be eligible to appear in external theory examination of the corresponding papers only.
- Any point regarding the examination in the above scheme, which is not covered, will be applicable as per the examination scheme of respective course declared by the University or M.P. Government, whichever may be applicable, and the final decision in this regard will be taken by the Principal on the recommendation of Examination Committee.

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T101 – CORE COURSE I – COMPUTER & COMMUNICATION FUNDAMENTALS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Computer Organization: Digital and Analog computers, Major components of a digital computer, Memory addressing capability of a CPU, Word length of a computer, Processing speed of a CPU, Definitions of Hardware, Software and Firmware. Definitions of Dumb, Smart and Intelligent terminals. Binary Systems: Digital Systems, Binary Numbers, Number Base Conversions, Octal and Hexadecimal Numbers, Complements, Signed Binary Numbers, Binary Codes: BCD code, Gray Code, ASCII code, Excess 3 Code, Error detecting Code. Computer Arithmetic: Binary representation of Negative Integers using 2's complement and Signed magnitude representation,	
Unit-II	**10 Lectures**
Boolean Algebra and Logic Gates: Basic Definitions, Basic Theorems and properties of Boolean Algebra, Boolean Functions, Canonical and standard forms, Other Logic operations, Digital Logic gates, Integrated Circuits. Gate-Level Minimization: The K-Map Method, 3 and 4 variable K-Map, Product of sums simplification, Sum of Products simplification, Don't care conditions, NAND and NOR implementations, Exclusive-OR function.	
Unit-III	**10 Lectures**
Combinational Logic: Combinational Circuits, Analysis Procedure, Design Procedure, Binary half adder, binary full adder, binary full subtractor, binary parallel adder, carry propagation delay and Propagation delay calculation of various digital circuits. Fast adder, Decimal Adder, Binary multiplier, magnitude comparator, Parity generator, seven segment display, BCD to excess three code converter, Decoders, Encoders, Multiplexers, and Demultiplexers Synchronous Sequential logic: Sequential circuits, Latches, Flip Flops: SR, D, JK, T. Master Slave JK Flip flop.	
Unit-IV	**10 Lectures**
Shift Registers- Serial in Serial out, Serial in Parallel out, Parallel in Serial out and Parallel in Parallel out. Designing of Asynchronous (Ripple) Counters, Design of Synchronous Counters. Types of communication with and among computers, Characteristics of communication channels, allocation of channel, signal power, amplitude and frequency modulation, space division multiplexing, frequency division multiplexing, Time division multiplexing.	
Unit-V	**08 Lectures**
Need for computer communication networks, Internet and world wide web: E-mail, File transfer, Remote login. Networking topologies. LAN, MAN, WAN, Ethernet LAN. Layer 2 and layer 3 switches, Wireless LAN. Switch hub and router. Wireless LAN, interconnecting networks, IP address and IP data grams, Internet and intranet; inter security, virtual private networks, future of Internet technology.	

TEXT BOOKS:

1. Digital Design by M. Morris Mano. Publication: PHI Eastern economy edition (Year-2001)
2. Computer Architecture By Nicholas Carter, Schaum Series Adaptation, end edition, 2011

REFERENCE BOOKS:

1. Data communications and networking By A. Forouzan Publication: TMH Third edition (Year- 2004)
2. Computer Fundamentals – Architecture and Organization By B. Ram
3. Computer networks by Andrew Tanenbaum Publication: PHI Fourth edition (Year- 2003)
4. Computer organization and architecture by William Stallings

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T101 – CORE COURSE I – COMPUTER & COMMUNICATION FUNDAMENTALS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

1. Write the various units of CPU. Describe the working of each unit.
2. Write the definition of hardware, software and firmware.
3. What you mean by word length of a computer.
4. If address bus for a CPU is 20 bits wide, then calculate the size of memory that can be attached to it.
5. If data bus for a CPU is 16 bit then what will be its word length.
6. What are the major factors that decide the processing speed of a CPU?
7. Write the first 20 decimal digits in base 4.
8. Write the first 20 decimal digits in base 3.
9. Add and multiply the following numbers in the given base without converting to decimal.
   i. (1230) 4 and (33)4 **ii.** (130) 5 and (34)5 **iii.** (230) 6 and (54)6 **iv.** (130.4) 5 and (34.4)5
10. Write the first 100 decimal digits into binary.
11. Convert the following numbers into binary.
    i. 123.56 **ii.** 456.75 **iii.** 345.9 **iv.** 890.9
    v. 567.9 **vi.** 668.7
12. Converts the following numbers into decimal.
    i. 10101010 **ii.** 101010110011 **iii.** 10110101.1111 **iv.** 101010111.1101
13. Perform the following conversion , without converting into decimal :
    i. (3674)8 to ( )16 to ( ) 2 **ii.** (1001010101010)2 to ( ) 16 to ( ) 8
    iii. (AC4)16 to ( )2 to ( ) 8
14. Simplify the following Boolean functions to minimum number of literals:
    i. AB+AB'+C'+ABC **ii.** ABC+ABC'+AB
    iii. AC+BC+ABC+BC' **iv.** A'CD + A'C'D' + A'B'C'D +ABC' +ABCD +A'B'C'D'
15. Simplify the following functions using karnaugh map.
    i. F=□?(1,4, 7, 8) **ii.** F=□?(3,4, 7) **iii.** F=□?(0,1,2,4, 7, 8,10,15,)
    iv. F=□?(1,4, 7, 8,10) and D=□?(2,11,12) **v.** F=□?(1,2,3,4,7,8,9,11,12,15)
16. Implement the following function using AND and OR gate.
    i. F=□?(1,4, 7, 8) **ii.** F=□?(3,4, 7)
    iii. F=□?(0,1,2,4, 7, 8,10,15,) **iv.** F=□?(1,2,3,4,7,8,9,11,12,15)
17. Implement the following function using only NOR gate.
    i. AB+AB'+C'+ABC **ii.** ABC+ABC'+AB **iii.** AC+BC+ABC+BC' **iv.** ABC'+BC+AB
18. Design a combinational circuit that accepts a three-bit number and generates an output binary number equal to the square of the input number.
19. Design a combinational circuit that accept BCD values and generate cube of that number.
20. A combinational circuit is defined by the following two functions.
    i. F1 = x'y'+xyz' **ii.** F2 = x'+y **iii.** F3 = xy+x'y'
21. Design an even parity generator.
22. Design 3 to 8 decoder by using two 2 to 4 decoder.
23. Design 4 to 16 decoder by using two 3 to 8 decoder.
24. Design 5 to 32 decoder by using four 3 to 8 decoder.

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T102 – CORE COURSE II – PROGRAMMING & PROBLEM SOLVING USING C

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**14 Lectures**
Introduction to Computer based Problem Solving, Classification of programming languages, Programming Environment {Assemblers, compilers, interpreters, linkers, and loaders}. Features of good programs – Structured program, modular program, fundamentals of programming. Classification of programming languages: low-level language, middle-level language, high -level language along with examples and differences. Programming Concepts with Flowcharts and algorithms, How to make flowchart: Introduction of the symbols of flowchart and algorithm. Importance of Flowchart and algorithms. Solving and discussing problems with flowchart and algorithm. Class assignment on flowchart.	
Unit-II	**14 Lectures**
Overview of C: C character set, identifiers and keywords. Data types (description of different types of data types along with examples.), Format Specifiers associated with each data type. Constants and Variables: Some examples of constants and variables. Expressions and Operators: Instructions, declaration, arithmetic, unary, relational, logical, assignment and conditional operators. Sequence Control Structures, Decision Control Structures (ifelse and nested if-else) along with examples. Some exercises based on decision control statements. Loop Control Structures: Introduction of loops like for loop and some examples of "for loop". Examples of do-while loop, while loop, break, continue. Explanation and examples of Case Control Structures. Nesting of loops i.e. loops in a loop along with examples. Introduction of Arrays: Array initialization. Programs using 1D Array. Advantages and disadvantages of Arrays. 2D and multidimensional array and programs based on it.	
Unit-III	**14 Lectures**
Introduction of Pointers: Pointers declaration, pointer arithmetic and operation. Concept of Functions: prototype and parameter passing and some programs based on it. Advantages/pitfalls of pointers. Dynamic Memory allocation: malloc() and calloc() functions. Pointers and their Applications: String Handling. Call by reference and Call by value. String Handling: Pointers and strings, standard library string functions. Introduction of Storage classes (register, static, auto and extern) and their differences. Introduction of Structures: Structure declaration, program and application. Introduction of Union: Union declaration, program and application. Difference between Structure and Union. Some standard Functions like gets(), goto(), puts(), getchar(), putchar(), getche() etc.	
Unit-IV	**12 Lectures**
Introduction of user defined functions along with examples. Functions with arguments, without arguments along with the examples. Communication between functions (how one function calls the other functions). Some exercise based on functions. Command line Arguments. Enumerations and bit fields and program based on it. Introduction of Recursion and program based on it.	
Unit-V	**10 Lectures**
Concept of Pre-Processors: macro expansion, file inclusion Conditional compilation. Macros with argument and macro versus function. Introduction to file handling: Different operations on file like read write and append. File Creation and programs based on file creation, reading and merging.	

Text Book:

1. Let us C, Yashwant Kanetkar, BPB Publications

REFERENCE BOOKS:

1. B.W. Kernighan & D.M. Ritchie, "The C Programming Language", Prentice Hall of India
2. Ashok N. Kamthane , "Programming with ANSI and Turbo C", Pearson Education
3. Ashok N. Kamthane et. al., Computer Programming and IT (for RTU), Pearson Education, 2011

Syllabus
M.Sc. (C.S.). Part I – Semester I

MCS - T102 – CORE COURSE II – PROGRAMMING & PROBLEM SOLVING USING C

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

Class Assignments:

1. Write a C program to display "Hello Computer" on the screen.
2. Write a C program to display Your Name, Address and City in different lines.
3. Write a C program to find the area of a circle using the formula: Area = PI * r2.
4. Write a C program to find the area and volume of sphere. Formulas are: Area =4*PI*R*R Volume = 4/3*PI*R*R*R.
5. Write a C program to print the multiply value of two accepted numbers.
6. Write a C program to read in a three digit number produce following output (assuming that the input is 347)
   3 hundreds 4 tens 7 units
7. Write a C program to read in two integer values and display one as a percentage of the other. Typically your output should look like 20 is 50.00% of 40 assuming that the input numbers were 20 and 40. Display the percentage correct to 2 decimal places.
8. Write a C program to find out whether the character presses through the keyboard is a digit or not (using conditional operator).
9. Write a C program to swap variable values of i and j.
10. Write a C program to find the maximum from given three nos.
11. Write a C program to find that the accepted no. is Negative, Positive or Zero.
12. Write a program which reads two integer values. If the first is lesser print the message up. If the second is lesser, print the message down if they are equal, print the message equal if there is an error reading the data, print a message containing the word Error.
13. Write a C program that prints the given three integers in ascending order using if –else.
14. Given as input three integers representing a date as day, month, year, print the number day, month and year for the next day's date. Typical input: "28 2 1992" Typical output: "Date following 28:02:1992 is 29:02:1992".
15. Write a C program for calculator designing using switch /case loop.
16. Write a C program to convert decimal to binary.
17. Write a C program to convert decimal to hexadecimal.
18. Write a C program to display first 25 Fibonacci nos.
19. Write a C program to display first 100 prime nos.
20. Write a C program to find factorial of accepted no.
21. Write a C program to find the sum of digits of accepted no.
22. Write a C program to print the accepted no and its reverse no.
23. Write a C program to print all the factors of accepted no.
24. Write a C program to find HCF of two given numbers.
25. Write a C program to find LCM of two given numbers.
26. Write a C program to find all the prime numbers between two given numbers.
27. Write C programs to print the terms of each of the following series: ii. iii. iv. v. vi.
    i. Sin(x) ii. Cos(x) iii. Log (1+x) iv. log (1-x) v. e^x vi. e^{-x}
28. Write a C program to print the sum of series. (Will be given in class)
29. Write a C program to find minimum, maximum, sum and average of the given one dimensional array.
30. Write a C program to perform the basic Matrix operations addition, subtraction, multiplication, Transpose.
31. Write a program to take a sentence as input and reverse every word of the sentence.
32. Write a C Function for the following task:
    a) Calculating Factorial
    b) Nth Fibonacci number
    c) Swapping of two variables
    d) Minimum/maximum value from the given input values.

Syllabus
M.Sc. (C.S.). Part I – Semester I

MCS - T102 – CORE COURSE II – PROGRAMMING & PROBLEM SOLVING USING C

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

Class Assignments:

33. Display the following output on screen (assuming the value for input parameter n=5) :

a. * ** *** **** *****	b. 1 12 123 1234 12345	c. A AB ABC ABCD ABCDE	d. 1 23 345 4567 56789	e. 1 23 456 78910 01112131415
f. ***** **** *** ** *	g. ABCDE ABCD ABC AB A	h. * *** ***** ******* *********	I. 1 123 12345 1234567 123456789	j. 1 121 12321 1234321 123454321
k. 1 01 101 0101 10101	l. 1 22 333 4444 55555	m. A AB ABC AB A	n. ABCDEDCBA ABCD DCBA ABC CBA AB BA A A	o. 1 121 12321 1234321 123454321

34. Write User Defined Function and test them in the main program for the following standard function:

a) int mystrlen(char *s)
b) char *mysubstr(char *s, int i, int j)
c) char *mystrcat(char *s1,char *s2)
d) int mystrcmp(char *s1, char *s2)
e) char *mystrrev(char *s)
f) int mystrend(char *s, char *t)
g) char *myreplace(char *s, char *old, char *new)
h) char *mytoupper(char *)
i) char *mytolower(char*)

35. Write the following recursive C Function:

a) Factorial of a given number
b) Nth Fibonacci number
c) Reverse of a given String
d) Reverse of a give Number

36. Write a c program to create a new data type Date with the help of structure and typedef. Also write following user defined function for date manipulation.

a) To return next Date
b) To return next Month
c) To return next Year
d) To add few Days in a date
e) To add few Months in a date
f) To return the date of the week of a given date
g) To return Month name from the date
h) To Display the Date in various format as: Date Display (Date d1, char *format) Here Date is the newly created data type. The format string can hold the following values: "DDMMYYYY", "MMDDYY", "MON, DD, YYYY".

37. Write a C program to implement myprintf and myscanf functions using concept of variable number of arguments. (using getch, putch, gets and puts function).

38. Write a C program that creates an Employee text file. Records are empid, empname, designation, qualification, salary, experience, Research work, address, city phone.

39. Write C programs for the following operation to work like DOS Commands:

a) Type abc.txt
b) Copy source1.txt source2.txt
c) Compare source1.txt source2.txt
d) Concatenate source1.txt source2.txt

40. Write a C program to open two files containing integers (in sorted order) and merge their contents.

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T102 – CORE COURSE II – PROGRAMMING & PROBLEM SOLVING USING C

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

Class Assignments:

41. Write a C program to count the number of vowels, consonants, digits, spaces, other symbols, words and lines in a given text file.

42. Write C code to check if an integer is a power of 2 or not.

43. Write a C program to set a particular bit in a given number.

44. Write a C program to reset a particular bit in a given number.

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T103 – CORE COURSE III – OPERATING SYSTEMS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**13 Lectures**
Definition, function and goal of Operating system. Processor Management: Concepts, state, PCB, Scheduling Queue, Schedulers, context switching. CPU Scheduling Algorithms: FCFS, SJF, SRTN, RR, Priority scheduling, Multiple queue.	
Unit-II	**14 Lectures**
Memory Management: Overlays, Swapping, Logical and Physical address space. Continues memory allocation: fragmentation, paging, segmentation. Virtual memory management: Introduction, demand paging, page replacement algorithm (FIFO, Optimal, and LRU), thrashing.	
Unit-III	**13 Lectures**
Concurrent Processes: interprocess communication, interprocess synchronization, semaphores, classical problem in concurrent programming. Deadlock: characteristics, resource allocation graph, deadlock prevention, deadlock avoidance (Banker algorithm), deadlock detection and recovery.	
Unit-IV	**12 Lectures**
File Management: operations on a file, structure of a file System. Directory implementation: allocation method, free space management, sharing and protection. UNIX file system, device management: Goals of input/output software design.	
Unit-V	**12 Lectures**
Layers of I/O software, structure of device drivers, disk arm scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN, and LOOK). Introduction to network and distributed operating systems.	

BOOKS:

1. Operating System Concepts, Addison Wesley, 4th Edition, A. Silberschatz and P. Galvin. 1994

REFERENCE BOOKS:

1. Design of Unix operating system, Bach M., Pearson Education
2. Operating systems, 4rth Edition, William Stallings, Pearson Education, 2003

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T104 – CORE COURSE IV – DISCRETE STRUCTURES

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week : 04 Hours **Total Lectures: 64**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**12 Lectures**
The Foundations: Logic, Sets and Functions: Introduction to set theory, mathematical logic, prepositions, prepositional equivalences, predicates and quantifiers. Fuzzy sets, functions. **Mathematical reasoning:** Mathematical induction. Use of Mathematical induction to solve different problems.	
Unit-II	**12 Lectures**
Combinatorics: The basics of counting, The sum rule and The product rule, The Pigeonhole Principle, Permutations with repetitions, Permutations without repetitions, Circular Permutations. Recurrence relations, solving recurrence relations..	
Unit-III	**13 Lectures**
Relations: Relation definition , Relations and their properties, Unary relations , Binary relations, Ternary relations , equivalence relations, partial ordering. Operations on relations – union, intersection and join. Concepts of least upper bond, Greatest lower bond, maximal element, minimal element, Greatest element, Least element of a partially ordered set, lattices, sub lattices,	
Unit-IV	**13 Lectures**
Graphs: Introduction to Graphs, Importance of graph theory in computer science, Graph terminology, representing graphs, graph types, graph isomorphism. Connectivity, Euler and Hamiltonian Paths, shortest path problems, planar graphs, graph colouring, chromatic number, Euler's formula. Applications of Graph Colouring, Introduction to Trees, applications of trees, tree traversal, trees and sorting, Spanning trees, minimum spanning trees.	
Unit-V	**14 Lectures**
Languages and Grammars: Introduction to Languages and Grammars, solving problems for validity of statements according to the grammar. Importance of Language theory in Computer Science, Importance of Derivation trees, Importance of Parsing, Types of Phrase structure grammars.	

TEXT BOOK:

1. Kenneth H. Rosen "Discrete Mathematics and its Applications", 5th edition, Tata McGraw-Hill Edition.
2. Let us C, Yashavant Kanetkar, BPB Publications

REFERENCE BOOKS:

1. Kolman, Busby & Ross "Discrete Mathematical Structures", 5th edition Pearson Education
2. Narsingh Deo "Graph Theory with Applications to Engineering. & Computer Science", 4th edition, Prentice Hall of India
3. Discrete Structures, Logic and Computability by James L. Hein, 2nd edition, Narosa Publishing House
4. Discrete and Combinatorial Mathematics, 5th Edition by Ralph P. Grimaldi, Addison-Wesley Publication

Class Assignments:	
1.	Find a formula for the sum of the first *n* even positive integers.
2.	Use mathematical induction to prove the formula that you found in Exercise 1.
3.	Use mathematical induction to prove that 3 + 3 • 5 +
4.	5**2** +-----+ 3-5" = 3(5"+' - l)/4 whenever *n* is a nonnegative integer.
5.	Use mathematical induction to prove that 2 — 2 . 7 + 2.7**2**------+ 2(-7)" = (1 - (-7)"+l)/4 whenever *n* is a nonnegative integer.

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T104 – CORE COURSE IV – DISCRETE STRUCTURES

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week : 04 Hours **Total Lectures: 64**

Class Assignments:

6. Find a formula for ½ +1/4 + 1/8 +…….+1/2**n** by examining the values of this expression for small values of n. Use mathematical induction to prove your result.

7. Show that l2 + 22 + • • • + n2= n(n + l)(2n + l)/6 whenever *n* is a positive integer.

8. Show that l3 + 23 + • • • + *n3 = [n(n +* l)/2]2 whenever *n* is a positive integer.

9. Prove that l2 + 32 + 52 + ….. + (2n + l)2 = (n + 1) *(n* + l)(2n + 3)/3 whenever *n* is a nonnegative integer.

10. Prove that 1 • 1! + 2 . 2! +----+ *-n . n! = (n +1)! – 1* whenever *n* is a positive integer.

11. Show by mathematical induction that if *h* > — l , then 1 + *nh* <= (1 + *h)*n for all nonnegative integers *n.* This is called **Bernoulli's inequality.**

12. Prove that 3" < *n!* whenever *n* is a positive integer greater than 6.

13. Show that *2n* > *n2* whenever *n* is an integer greater than 4.

14. Use mathematical induction to prove that ***n! < n*** **n** whenever *n* is a positive integer greater than 1.

15. Prove using mathematical induction that 1 -2 + 2-3 +……..+ n(n+1) = *n(n* + l)(n+2)/3.

16. Whenever n is a positive integer.

17. Use mathematical induction to show that 3 divides n **3** + *2n* whenever *n* is a nonnegative integer.

18. Use mathematical induction to show that 5 divides *n* **5** — *n* whenever *n* is a nonnegative integer.

19. Use mathematical induction to show that 6 divides *n* **3** — *n* whenever *n* is a nonnegative integer.

20. Use mathematical induction to show that ***n2*** — 1 is divisible by 8 whenever *n* is an odd positive integer.

21. Use mathematical induction to show that ***n2*** — *7n +* 12 is nonnegative if *n* is an integer greater than 3.

22. Use mathematical induction to prove that a set with *n* elements has *n(n* — l)/2 subsets containing exactly two elements whenever *n* is an integer greater than or equal to 2.

23. Use mathematical induction to prove that a set with *n* elements has *n(n* — \)*(n* — 2)/6 subsets containing exactly three elements whenever *n* is an integer greater than or equal to 3.

24. Write a small procedure for addition of a node in a binary search tree.

25. Define the following terms :-

a) Graph Isomorphism b) Map Coloring c) Chromatic Polynomial d) Phrase Structure grammar
e) Euler graph f) Bipartite Graph g) Hamiltonian circuits

26. Write a small C code to create a digraph.

27. Write a small C code for insertion of a node in binary search tree.

28. Implement Topological sorting algorithm in C language.

29. Write a C code to decide whether a graph is Bipartite.

30. Write a small C code to create a graph and find its adjacency matrix.

31. Write a small C code to create a graph and find its adjacency list.Write a small C code to create a graph and find its incidence matrix.

32. Implement Dijstra's algorithm in C.

33. Write a small C code to determine isomorphism of two graphs.

34. Write a small C code to determine whether a graph is euler graph.

35. Write a small C code to determine whether a graph has Hamilton path.

36. Implement Prim's algorithm in C for finding minimum spanning tree.

37. Implement Kruskal's algorithm in C for finding minimum spanning tree.

38. Write a small C code to create a graph and find its subgraphs.

39. Write a small C code for breadth first traversal.

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - T105 – CORE COURSE V – COMMUNICATION SKILLS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 02 Hours **Total Lectures: 32**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**06 Lectures**
Fundaments of Communication (OHP & PPP): Definition, Importance, Forms of Communication, Process of Communication, Channels, Barriers and Strategies, How to overcome Barriers of Communication.	
Unit-II	**06 Lectures**
Listening (PPP): Definition, Importance, Benefits, Barriers, Approaches, Be a Better Listener, Exercises and Cases.	
Unit-III	**06 Lectures**
Group Discussions :(PPP) Definition, Importance, Process, Points to be borne in mind while participating in a Group Discussion, Dos and Don'ts of Group Discussion **Interview (PPP)** Types of Interviews, Points to be borne in mind as an interviewer and an Interviewee, Commonly asked Questions, Dos and Don'ts.	
Unit-IV	**08 Lectures**
Transactional Analysis: (PPP) Transactional Analysis, Johari Window, (PPP) **Written Communication:** Cover letter, Report Writing, Business Correspondence, Preparation of Project Reports, Drafting emails.	
Unit-V	**06 Lectures**
Negotiation Skills: Basic Principles, Process of Negotiation, Essentials of Negotiation. Contemporary Communication Styles, Technology Enabled Communication	

TEXT BOOK:

1. Chaturvedi, P.D. and Chaturvedi Mukesh (2004), "Business Communication" (Pearson Education, Singapore Pvt. Ltd.)
2. Fred Luthans, Organizational Behaviour

REFERENCE BOOKS:

1. Business Communication by ICMR, Feb 2001
2. Heller Robert (1998), "Essential DK Managers: Communication Clearly", Dorling Kindersley, London

CLASS ASSIGNMENTS:

1. Preparing Resume
2. Writing Formal letters, Memos, Drafting emails, Notices
3. Create a Questionnaire and meet people whom you have never met before and know about them

Syllabus

M.Sc. (C.S.). Part I – Semester I

MCS - P106 – CORE COURSE VI

PRACTICAL COURSE IN OPERATING SYSTEMS

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and mathematical under the guidance of respective faculty.

MCS - P107 – CORE COURSE VII

PRACTICAL COURSE IN PROGRAMMING & PROBLEM SOLVING THROUGH C

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and mathematical under the guidance of respective faculty.

MCS – 108 (SKEG) – SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC) –

SKEG-T/P122 – PC SOFTWARE & INSTALLATIONS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

PART A : THEORY - MAX. MARKS: 50 + 30 **MIN. PASS MARKS: 20 + 12**

No. of Lectures per week : 02 Hours **Total Lectures: 32**

PART B : PRACTICALS - MAX. MARKS: 20 **MIN. PASS MARKS: 08**

No. of Laboratory per week: 02 Hours **Total Lectures: 32**

SKEG- T-119 – PERSONALITY DEVELOPMENT

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week: 03 Hours **Total Lectures: 48**

SKEG-T116– MANAGERIAL SKILLS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours **Total Lectures: 48**

SKEG-T108 – HEALTH EDUCATION

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours **Total Lectures: 48**

Syllabus
M.Sc. (C.S.). Part I – Semester II

MCS – T201 – CORE COURSE I– DATA STRUCTURES USING C++

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**14 Lectures**
Introduction to C++: Structures, Variables in C++, Functions, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions. **Class and Objects**: Introduction to Classes and Objects Constructors, destructors, friend function, dynamic memory allocation, Inheritance, Overloading, Polymorphism.	
Unit-II	**14 Lectures**
Definition of data structures and abstract data types. Examples and real life applications **Data Structures**: Arrays, Stacks, Queues, Dequeues	
Unit-III	**12 Lectures**
Linked Lists, Singly and Doubly linked list Applications of Linked Lists, Linked implementation of queue and stacks **Binary Search Tree:** Definition and Implementation- preorder, post order, inorder traversal, Red Black Tree Definition.	
Unit-IV	**12 Lectures**
Graphs: Definition and implementation, Hash function, Collision Resolution Techniques, Hashing Applications, Time Complexity, Big – Oh - notation, Running Times, Best Case, Worst Case, Average Case, Introduction to Recursion, Divide and Conquer Algorithm, Evaluating time Complexity	
Unit-V	**12 Lectures**
Linear or Sequential Search, Array implementations, Linked implementation, Binary Search, Interpolation Search. Introduction, Sorting by exchange, selection, insertions, Bubble sort, Selection sort, Insertion sort, Efficiency of above algorithms, Merge sort, Quick sort Algorithm, Heap sort, Heap Construction, Heap sort.	

TEXT BOOK:

1. Data Structures using C by A. M. Tenenbaum, Langsam, Moshe J. Augentem, PHI Publications

REFERENCE BOOKS:

1. Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original Edition, Addison- Wesley, 1999, Low Priced Edition
2. Fundamentals of Data structures by Ellis Horowitz & Sartaj Sahni, Pub, 1983, AW

Class Assignments:

1. Write a program to display message.
2. Write a program to find the area of circle, rectangle or triangle.
3. Write a program that allows the user to enter the number, and then generate the table, formatting it into ten columns and 20 lines. Interaction with the program should look like this. Enter a number: 2

2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
22	**24**	**26**	**28**	**30**	**32**	**34**	**36**	**38**	**40**
42	**44**	**46**	**48**	**50**	**52**	**54**	**56**	**58**	**60**

4. Write a temperature – conversion program that gives the user the option of converting Fahrenheit to Celsius or Celsius to Fahrenheit. Then carry out the conversion. Use Floating point numbers.

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T201 – CORE COURSE I– DATA STRUCTURES USING C++

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

Class Assignments:

5. Create the equivalent of four-function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation; adding, subtracting, multiplying or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user want to do another calculation. The response can be 'y' or 'n'.
6. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers.
7. A point on the two-dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis, and 5 units up the Y axis. The sum of two points can be defined as new points, and whose Y coordinate is the sum of their Y coordinates.
8. Write a program that uses a structure called point to model A point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point.
9. Create a structure called volume that uses three variables of type Distance to model the volume of a room. Initialize a variable of type Volume to specific dimensions, and then calculate the volume it represents, and print out the result. To calculate the volume, convert each dimension from a Distance variable to a variable of type float representing feet and fractions of a foot, and then multiply the resulting three numbers.
10. Write a function called circarea ( ) that finds the area of a circle in a similar way. It should take an argument of type float and return an argument of the same type. Write a main ( ) function that gets a radius value from the user, calls c l r c a r e a ( ), and displays the result.
11. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called p o w e r ( ) that takes d o u b l e value for n and an i nt value for p, and returns the result as d o u b l e value. Use a default argument of 2 for p, so that if this argument is committed, the number will be squared. Write a main ( ) function that gets values from the user to test this function.
12. Write a function called z e r o Smaller ( ) that is passed two int argument by reference and then sets the smaller of the two numbers to 0. Write a main ( ) program to exercise this function.
13. Write a function that takes two Distance values as arguments and returns the larger one. Include a main ( ) program that accepts two Distance figures from the user, compares them, and displays the larger.
14. Create a class that imitates part of the functionality of the basis data type int, Call the class int (note different spelling). The only data in this class is an int variable. Include member functions to initialize an Int to 0, to initialize it to an int value, to display it (it looks just like an int), and to add two Int values.
15. Write a program that exercises this class by creating two initialized and one un-initialized int values, adding these two initialized values and placing the response in the un-initialized value, and then displaying this result.
16. Create a class called time that has separate int member data for hours, minutes, and second. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it, in 11:59:59 format.

    The final member for Students Admitted in July, 2011 onwards function should add two objects of type time passed as arguments. A main ( ) program should create two initialized time objects, and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third time variable. Finally it should display the value of this third variable.

Syllabus
M.Sc. (C.S.). Part I – Semester II

MCS – T201 – CORE COURSE I– DATA STRUCTURES USING C++

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

Class Assignments:

17. Write a function called reversit ( ) that reverse a string (anarray of char). Use a for loop that swaps the first and last characters, then the second and next-to-last characters, and so on. The string should be passed to reverse it ( ) as an argument.Write a program to exercise reversit the program should get a string from the user, call reversit ( ) and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba".
18. Create a class called employee that contains a name (an array of char) and an employee number (type Long). Include a member function called get data ( ) to get data from the user for insertion into the object, and another function called put data ( ) to display the data. Assume the name has no embedded blanks.Write a main () program use this class. It should create an array of type employee, and then the user input data for up to 10 employees. Finally, it should print out the data for all the employees.
19. Write a program that substitutes an overloaded += operator for the overloaded+ operator in the STRPLUS program. This operator should allow statements likes1 +=s2; where s2 is added (concentrated) to s1 and the result left in s1. The operator should also permit the results of the operation to be used in other calculations, as in s3 = s1 +=s2.
20. Create a class int. Overload all five integer arithmetic operators (+; =; *; /; and %) so that they operate an objects of type int. If the result of any such arithmetic operation exceeds the normal range of ints = from = 32,768 to 32;767 – have the operator print a warning and terminate the program. Such a data type might be useful where mistakes caused by arithmetic overflow are unacceptable. Hint: To facilitate checking for overflow; perform the calculations using type long. Write a program to test this class.
21. Imagine a publishing company that markets both book and audio-cassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes; book, which adds a page count (type int); and tape, which adds a playing time in minutes (type float). Each of these three classes should have a get data () function to get its data from the user at the keyboard, and a put data ( ) function to display its data.Write a main ( ) program to test the book and tape classes by creating instances of them, asking the user to fill in their data with get data ( ), and then displaying the data with put data ( ).
22. Write a program that reads a group of numbers from the user and places them in an array of type float. Once the numbers are stored in the array, the program should average them and print the result. Use pointer notation wherever possible.
23. Start with the String class from the NEWSTR. Add a member function called up it ( ) that converts the string to all upper case. You can use the toupper ( ) library function, which takes a single character as an argument and return a character that has been converted (if necessary) to uppercase. This function uses the CTYPE,h H header file. Write some code in main ( ) to test this function.
24. Add a destructor to the LINKLIST program. It should delete all the links when a link list object is destroyed. It can do this by following along the chain, deleting each links as it goes. You can test the destructor by having it display a message each time it deletes a link; it should delete the same number of links that were added to the list. (A destructor is called automatically by the system for any existing objects when the program exits.)
25. Imagine the same publishing company that markets both book and audio-cassette versions of its works. As in that exercise, create a class called publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int); and tape, which adds a playing time in minutes (type Effective for Students Admitted in July, 2011 onwards float). Each of the three classes should have a get data ( ) function to get its data from the user at the keyboard, and a put data ( ) function to display the data.
26. Write a C++ Program to construct a Class stack of integers and to perform.
27. The following operations on it: a) Push b) Pop c) Display

Syllabus
M.Sc. (C.S.). Part I – Semester II

MCS – T201 – CORE COURSE I– DATA STRUCTURES USING C++

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

Class Assignments:

28. The program should print appropriate messages for stack overflow, stack underflow, and stack empty.

29. Write a C++ Program to simulate the working of a queue of integers.

30. Using an array. Provide the following operations: a) Insert b) Delete c) Display

31. Write a C++ Program to simulate the working of a circular queue of integers using an array. Provide the following operations: a) Insert b) Delete c) Display

32. Write a C++ Program to convert and print a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

33. Write a C++ Program to evaluate a valid suffix/postfix expression using stack. Assume that the suffix/postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).

34. Write a C++ Program using dynamic variables and pointers, to construct a singly linked list consisting of the following information in each node: student id (integer), student name (character string) and semester (integer). The operations to be supported are:

a) The insertion operation
   I. At the front of a list II. At the back of the list III. At any position in the list
b) Deleting a node based on student id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.
c) Searching a node based on student id and update the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.
d) Displaying all the nodes in the list.

35. Write a C++ Program using dynamic variables and pointers to construct a stack of integers using singly linked list and to perform the following operations: a) Push b) Pop c) Display

36. The program should print appropriate messages for stack overflow and stack empty.

37. Write a C++ program using dynamic variables and pointers to construct a queue of integers using singly linked list and to perform the following operations: a) Insert b) Delete c) Display

Effective for Students Admitted in July, 2011 onwards.

38. Write a C++ Program to support the following operations on a doubly linked list where each node consists of integers:

a) Create a doubly linked list by adding each node at the front.
b) Insert a new node to the left of the node whose key value is read as an input.
c) Delete the node of a given data, if it is found, otherwise display appropriate message.
d) Display the contents of the list.

39. Write a C++ Program:

a) To construct a binary search tree of integers. b) To traverse the tree using all the methods i.e., inorder.
c) Preorder and postorder. d) To display the elements in the tree.

40. Write a C++ program to implement multiway search trees.

41. Write a C++ program to implement AVL trees.

42. Write recursive C++ Programs for:

a) Searching an element on a given list of integers using the Binary Search method.
b) Solving the Towers of Hanoi problem.

43. Write a C++ program to implement Hashing Functions.

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T202 – CORE COURSE II – DATABASE MANAGEMENT SYSTEM

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Introduction: Advantages of DBMS approach, Various views of data, data independence, schema & sub-schema, primary concept of data models, database languages, transaction management, database administrator &user, data dictionary, database architectures. **ER model:** Basic concept, design issues, mapping constraint, keys, ER diagram, weak & strong entity-sets, specialization & generalization, aggregation, inheritance, design of ER schema, Reduction of ER Schema to tables. Domains, relation, kind of relation, Relational databases, Various types of keys: candidate, primary, alternate & foreign keys.	
Unit-II	**10 Lectures**
Relational Algebra and SQL: The structure, relational algebra with extended operations, modification of database, Idea of relational calculus. **Relational Database:** Basic structure of SQL, Set operation, Aggregate functions, Null values, Nested Sub queries, derived relations, views, Modification of database, join relation, Domain, relation & keys, DDL in SQL. **Programming concepts of PL/SQL, Stored procedure, Database connectivity with ODBC/JDBC.**	
Unit-III	**10 Lectures**
Functional dependencies: Basic definitions, Trivial & non trivial dependencies, closure set of dependencies & of attributes, Irreducible set of dependencies, FD diagram**. Normalization:** Introduction to normalization, non loss decomposition, First, second and third normal forms, dependency preservation, BCNF, multivalue dependencies and fourth normal form, join dependencies and fifth normal form. **Database Integrity:** general idea, integrity rules, Domain rules, Attributes rules, assertion, triggers, integrity & SQL	
Unit-IV	**10 Lectures**
Transaction Management: basic concept, ACID properties, transaction state, Implementation of atomicity & durability, Concurrent execution, Basic idea of serializability. **Concurrency & Recovery:** Basic idea of concurrency control, basic idea of deadlock, Failure Classification, storage structure-types, stable storage implementation, data access, recovery & Atomicity: log based recovery, deferred database modification, immediate database modification, checkpoints.	
Unit-V	**08 Lectures**
Storage Structure: overview of physical storage media, magnetic disk: performance & optimization, RAID. **File Organization:** File organization, Organization of records in files, basic concept of Indexing, ordered indices: B+ tree & B tree index files. **Query processing, Query optimization, Introduction to data mining and data warehousing.**	

TEXT BOOK:

1. Database System Concepts, Henry F. Korth, Tata McGraw Hill, 4th Edition

Reference Books:

1. Fundamentals of Database Systems, Elmasri R, Navathe S, Addison Wesley 4th Ed., ISBN 0321122267
2. An introduction to database system- Bipin C. Desai
3. An introduction to Database System - C.J Date
4. SQL, PL/SQL The programming language of Oracle- Ivan Bayross

Syllabus
M.Sc. (C.S.). Part I – Semester II

MCS – T202 – CORE COURSE II – DATABASE MANAGEMENT SYSTEM

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

(Student should submit their assignment in hard copy (hand written) as well as soft copy to the respective faculty)

A. Prepare the following case study on ER diagram and normalized database design based on FD's

(a) Database design for Retail Banking
(b) Database design for Technical Training Institute
(c) Database design for an Internet Book Shop
(d) Database design for Customer Order Warehouse
(e) Database design for University Registrar's Office

B. Define the schema for the following databases with specific data type and constraints, the table name and its fields name are to be taken from database description which are given below :

1. Sales Information System

A database is being constructed for storing sales information system. A product can be described with a unique product number, product name, selling price, manufacturer name. The product can be sold to a particular client and each client have it own unique client number, client name, client addresses, city, pin code, state and total balance to be required to paid. Each client orders to buy product from the salesman. In the order, it has unique sales order number, sales order date, client number, salesman number (unique), billed whole payment by the party or not and its delivery date. The information associated with salesman is name, addresses, city, pin code, state, salary of the sales man, delivery date, total quantity ordered, product rate. Write the SQL queries for the following –

a) Retrieve the list of names and the cities of all the clients
b) List the various products available.
c) Find the names of all clients having 'a' as the second letter in their names.
d) List all the clients who are located in 'INDORE'.
e) Find the products whose selling price is greater than 2000 and less than or equal to 5000.
f) Add a new column NEW_PRICE into the product master table.
g) Rename the column product rate of Sales_Order_Details to new_product_rate.
h) List the products in sorted order of their description.
i) Display the order number and date on which the clients placed their order.
j) Delete all the records having delivery date before 25th March, 2010.
k) Change the delivery date of order number ON01008 to 16-05-10.
l) Change the bal_due of client_no CN01003 to 1200.
m) Find the product with description as 'HDD1034' and 'DVDRW'.
n) List the names, city and state of the clients not in the state of 'MP'.
o) List of all orders that were cancelled in the month of March.

2. College Department Management

A student is described by a unique Roll Number, Name, Address, and Semester. Each student enrols himself/herself in an Academic programme offered by a Department. Academic programmes have programme name(unique), duration, a programme code(unique) and a list of subjects (both core and elective subject) while the departments have department code (unique), department name (unique), HoD who is a Teacher and list of courses offered by it. Each teacher is described by employee code (unique), name, department and designation. A student registers Effective for Students Admitted in July, 2011 onwards some courses in a semester. A course is described by a unique course number, title of the course, credit allotted for the course and offering department. Database stores the grades obtained by different students in different courses registered by him/her in different semesters. Database also stores information about the courses offered by a department in a semester, the corresponding teacher(s) for each course. Write the SQL queries for the following –

Syllabus
M.Sc. (C.S.). Part I – Semester II

MCS – T202 – CORE COURSE II – DATABASE MANAGEMENT SYSTEM

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

a) Find all the students' name, city, course allotted from the SCS department.
b) List the total number of Faculty in the SCS department.
c) List the available courses from the SCS department.
d) List the all students in a particular semester.
e) List the students who earned CGPA greater than or equal to 8.5.
f) How much subjects are registered by a student in each semester.
g) List the common students who are allotted the same courses of both the programme MCA and M. Tech.
h) List the total number of student enrolled in the subject DBMS.
i) Retrieve the semester of the student under DBMS subject.
j) Retrieve the entire student name and arrange into ascending order.
k) Modify a student address DEWAS to INDORE where sdt_id='CSI08002'.
l) Find the total credit point of student required to complete for a course like MCA.
m) List the all courses which are related to computer science.
n) Retrieve all the students located at 'INDORE'.
o) Find the total number of department in our database.
p) List the all courses which are related to computer science.

3. Bank Database System

A bank database keeps record of the details of customers, accounts, loans and transactions such as deposits or withdraws. Customer record includes customer id, customer name, address, age, contact number, email id etc., accounts details are account number, account type(fixed account, savings account, monthly account etc), date of creation of the account. Transaction detail keeps information about amount deposited or withdrawn to/from a particular account and the date of transaction. The database should also store record of loans which include loan amount, loan date and the account number to which the loan is granted. Make appropriate tables for the above database and try to find out the following queries:

a) List the details of account holders who have a 'savings' account.
b) List the Name and address of account holders with loan amount more than 50,000.
c) Change the name of the customer to 'ABC' whose account number is 'TU001'.
d) List the account number with total deposit more than 80,000.
e) List the number of fixed deposit accounts in the bank.
f) Display the details of customers who created their accounts between '20-jan-10' to '20- Mar-10'.
g) Display the detailed transactions on 28th Feb, 2010.
h) Display the total amount deposited and withdrawn on 29th Aug, 2008.
i) List the details of customers who have a loan.

4. Library Information System

A library database stores information about books, journals, magazines etc. Searching for books can be done by author, title, subjects etc. Similarly journals can be searched by subject area, publisher etc. It should also be possible to see which book is issued to which student and belonging department.

Write the queries for the following –

a) List the names of the books issued between 21-Jan-10 and 29-Jan-10.
b) Retrieve the name and number of books by a particular author.
c) Retrieve the name of the publisher which has maximum number of books.
d) Count the total number of books in the library.
e) Count the number of books issued to a student with Roll no 'CSB06001'.
f) Change the author of the book to 'ABC' with book id='BK003'.

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T202 – CORE COURSE II – DATABASE MANAGEMENT SYSTEM

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

g) Retrieve the name of the student to whom the book named 'Database System', by 'E. Navathe' is issued.
h) Display the total number of books issued to different departments.
i) List the name of the books where subject is like 'ora'.

5. Hospital Information System

Hospital information system stores following information:
Patients - indoor/outdoor, medicines/lab tests (including results) prescribed to patients, information if a patient if referred to other expert/hospital. Doctors - specialization, patients attended etc. Different wards/beds and patients allotted to them etc. Patient registration form contains Registration number, Patient name, Address, Gender, Bed number, date of registration, refer doctor id etc. Doctor information contains Doctor code, Doctor Name, Specialization etc. Lab test information contains Test name, test number, test date, results and referred doctor's code. Bed information contains bed number, ward number and status (whether allotted or not). Queries:

a) Display the details of patients admitted between '20-jan-10' and '20-Mar-10'.
b) Change the name of the patient to 'Ram' whose patient id='PT011'.
c) Display the names of the patients and lab test results performed on '20-jan-10'.
d) Display the number of patients taking treatment under doctor ='ABC'.
e) Retrieve the name of doctor who is taking care of maximum number of patients.
f) Change the bed number of the patient to 456 where patient id='PT023'.
g) Change the status of bed with bed number 123 with 'not allotted'.
h) List the bed details which are free in ward number 10.
i) List the name of male patients in ward no 13 taking treatment under doctor 'XYZ'.
j) List the details of patients with age more than 50 taking treatment under a doctor, whose name like 'das'.

6. Payroll System of DAVV University

Write the queries for the following –

a) List all the employees of SCS department.
b) Retrieve all the employees who have the gross salary greater than or equal to Rs-21,000.
c) Find the DA, TA, HRA of the employee name 'Nitin'.
d) Find the total leave get by an employee.
e) List the employees department.
f) List the name grades of the employee of the SCS department.
g) Find how much tax will be required to pay by an employee in each month.
h) List the employees who are joined between '10-APR-07' to '28-AUG-08'.
i) Retrieve the mail id of all employees of SCS department.
j) Total leave available of the employee 'Ravindra'.
k) Add a new column 'employee phone' to a table employee.
l) List the employees with basic salary 8000.
m) Find the employees who have the highest bank balance.
n) Retrieve the employees who have the bank loan.
o) Find the bank balance of the employee 'SCSMP07001'.

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T203 – CORE COURSE III – SOFTWARE ENGINEERING

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Introduction to Software Engineering & Software Processes Software problem, Software engineering problem, Software engineering approach, Software characteristics and Applications. Software processes and its components, characteristics of software processes, Software development processes: Linear Sequential model, Prototyping model, RAD model, Iterative Enhancement model, Spiral model, Comparative study of various development models.	
Unit-II	**12 Lectures**
Project management process & Project Planning The people, product, process and project, Phases of project management process, Project life cycle, the W5HH principle. Software configuration management process, Management Process: Capability Maturity Model (CMM), Project Planning: Matrices and Measurement. Project estimation (Size & Cost), Project Scheduling, Staffing and personnel planning, Software configuration management plans, Quality assurance plans, Project monitoring plans, Risk management.	
Unit-III	**10 Lectures**
Software Requirement Analysis and Specification: Software requirements, Functional & Non functional requirements, Problem analysis (Structured analysis and Object Oriented analysis, Prototyping approach). Software Requirements specifications (SRS), Validation and Verification, Metrics. **Software Design:** Design principles: Problem partitioning and hierarchy, Abstraction, Modularity, Top-down and Bottom-up strategies. Effective Modular design: functional independency, Cohesion, Coupling. Structured design methodology.	
Unit-IV	**08 Lectures**
Software Quality Assurance: Quality concept, Quality management system, movements and assurance, Software reviews: formal and technical. Formal approaches to SQA, Software reliability, Capability Maturity Model (CMM), ISO 9000, Six sigma, SQA plan.	
Unit-V	**08 Lectures**
Software Testing: Software testing techniques: Testing fundamentals, White box testing, Black box testing, Testing for specialized environments, architectures and applications. Software testing strategies: A strategic approach to software testing, Strategic issues, Unit testing, Integration testing, Validation testing and system testing, The art of debugging.	

TEXT BOOKS:

1. An Integrated Approach to Software Engineering- Pankaj Jalote, Narosa Publishing House.
2. Software Engineering: Concepts & Practices – Dr. Ugrasen Suman, Cengage Learning, 2013.

REFERENCE BOOKS:

1. Software Engineering- Ian Sommerville, Pearson Education, New Delhi
2. Software Engineering Concepts-Richard E. Fairly, Tata McGraw Hill Inc. New York
3. Software Engineering: Principle & Practice-W. S. Jawadekar, Tata McGraw-Hill, New York

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T203 – CORE COURSE III – SOFTWARE ENGINEERING

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

Assignment 1:

Select a project that you want to develop and define the aim, objectives, and goals of your project. Select a software development methodology to develop the same project. Perform various phases of the methodology to produce work-products.

Assignment 2:

1. Suppose a program for solving a problem costs C, and a programming product for solving that problem costs 9C. Where do you think the extra 8C cost is spent? Suggest a possible breakdown of this extra cost.
2. If you are given extra time to improve the reliability of the final product developing a software product, how would you distribute this extra time?
3. Suggest ways of reducing the cost due to rework.
4. Suggest some ways to detect software errors in the early phases (when implementation is not yet complete.

Assignment 3:

1. Define & discuss Time boxing and Component based development model.
2. What problems will a Software Development Organization face if it does not have sufficiently document in its software process?
3. Suppose you were to plan to undertake the development of product base with large number of technical as well as customer, justify which life cycle model would you adopt.
4. Suggest a suitable life cycle model, for software project which your organization has undertaken on behalf of a certain change his requirements frequently justify.
5. Which of the following process models discussed in this chapter would you follow for the following projects? Give justifications.
    a) A simple data processing project.
    b) A data entry system for office staff who have never used computers before. The user interface and user friendliness are extremely important.
    c) A new system for comparing fingerprints. It is not clear if the current algorithms can compare fingerprints in the given response time constraints.
    d) A spreadsheet system that has some basic features and many other desirable features that use these basic features.
    e) A new missile tracking system. It is not known if the current hardware /software technology is mature enough to achieve the goals.
    f) An on-line inventory management system for an automobile industry.
    g) A flight control system with extremely high reliability. There are many potential hazards with such a system.
6. It is reasonable to assume that if software is easy to test, it will be easy to maintain. Suppose that by putting extra effort in design and coding you increase the cost of these phases by 15%, but you reduce the cost of testing and maintenance by 5%. Will you put in the extra effort?
7. Suppose you can measure the number the defects detected during the various reviews and testing. However, the customer requires an estimate of the Number of defects remaining at Effective for Students Admitted in July, 2011 onwards delivery time. How will you build a model to predict this? Assume the existence of any data you need.

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T203 – CORE COURSE III – SOFTWARE ENGINEERING

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

Assignment 4:

1. A database system is to be developed. After the requirements, its size is estimated to be 10,000 lines of code. Estimate the overall cost using the Watson and Felix model.
2. Consider a project to develop a full-screen editor. The major components identified are
   (1) screen edit (2) command language interpreter (3) file input and output
   (4) cursor movement (5) screen movement
   The sizes for these are estimated to be 4K, 2K, 1K, 2K and 3k delivered source code lines. Use the COCOMO model to determine
   (a) overall cost and schedule estimates (assumes values for different from 1.0).
   (b) cost and schedule estimates for different phases.
   (c) detailed cost and schedule estimates for the different components.
3. Suppose each communication path between two people consumes 5% of each person's time. For a project that requires 12 staff-months of programming work, how many people will be needed to finish the project in four months if
   (a) The democratic team structure is used.
   (b) The chief-programmer team structure is used? If the team consists of four persons, what is the difference in the completion time for a team using the democratic structure and a team using the chief-programmer structure?
4. Assume that testing (and bug fixing) effort is proportional to the number of errors detected (regardless of the nature of errors). Suppose that testing detects 90% of the total errors in the SW (10% remain undetected). By adding design and code reviews, suppose the cost of the design and coding phases increases by 10% each (from the base distribution given earlier), and 10% of the errors are detected in design reviews and 10% in code reviews. (So, testing now detects only 70% of errors). What is the impact on the overall cost of reviews?

Assignment 5:

1. Suppose you are developing a project in an organic mode. You have estimated the size of the product is about 100000 LOC. Compute normal effort and development time.
2. Suppose that an on-the-shelf s/w product for business applicant coast 15000/- and that its size is 40 KLOC. Assuming that in house engineer coast 6000/- per program month including overheads, would it be more effective by the product build it, which element of the cost are not included in the COCOMO MODEL. What additional factor should be considered making the buy and build decision.
3. The industry average productivity figure for engineers is only 10 LOC per day. What is the reason for the low productivity? Can be attributing the poor programming skills of the Software engineers.
4. As a project manager identify the traits that you would look for software engineers .While trying to select person for your team.
5. For the same number of lines of code and same development team size rank the following software project, in order of their estimated development time
   a) A text editor b) An employee payroll system
   c) An operating system for a new computer

Assignment 6:

1. What are central problems in producing the RS for a system?
2. Construct an example of an inconsistent (incomplete) SRS?

Syllabus
M.Sc. (C.S.). Part I – Semester II

MCS – T203 – CORE COURSE III – SOFTWARE ENGINEERING

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:

3. Make a friend of yours as the client. Perform structured analysis and object oriented analysis for the following problems:
(1) An electronic mail system
(2) A simple student registration system
(3) A system to analyse a person's diet
(4) A system to manage recipes for a household
(5) A system to fill tax forms for the current year tax laws

4. A library database contains entries that have the name of the book, followed by the author's name, the publisher's name, and year of publication, ISBN number of the book, and number of copies of the book. Each of the data entries is on a new line. Represent this database as a regular expression.

Assignment 7:

1. What is the cohesion of the following module? How would you change the module to increase cohesion?
Procedure file(file_ptr, filr_name, op_name)
Begin
Case op_name of
"open" perform activities for opening the file.
"close" perfor activities for opening the file.
"print" print the file.
End case
End.

2. What is the difference between a flow chart and a structure chart?

3. Draw the structure chart for the following program:

```
Main();
{
Int x, y;
X=0; y=0;
A(); b();}
A()
{x=x+y; y=y+5;}
B()
{x=x+5; y=y+x; a();}
```

How would you modify this program to improve the modularity?

4. Use the structured design methodology to produce a design for the following :
i. A system to convert ASCII to EBSDIC?
ii. A system that acts as a calculator with only basic arithmetic functions?
iii. A system does student registration in the manner it is done at your college?
iv. A system to manage the inventory at a hardware store?

5. Given a structure with high fan-out, how would you convert it to a structure with a low fanout?

Assignment 8:

1. Define error, fault, and failure? What is the difference between a fault and a failure? Does testing observe faults or failures?

2. What are the different levels of testing and the goals of different levels?

3. What is the goal of testing? Why the psychology of tester is important?

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T204 – CORE COURSE IV – COMPUTER ARCHITECTURE AND ORGANIZATION

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**14 Lectures**
Technological trends, measuring performance, speed up Amdahl's law. Basic structure of computer hardware: Functional units and components in computer organization: The memory unit, the input and output subsystem, the bus structures, ALU. Program development tools: Compiler, interpreter, and assembler. Introduction to 8085 micro processor architecture. pin out diagram of 8085 microprocessor. Addressing modes of 8085 processor.	
Unit-II	**14 Lectures**
Instruction set of 8085 processor: data transfer instructions, arithmetic instructions, branch instructions etc. assembly language programming examples. Introduction to 8088 microprocessor. Bit pair multiplication. Integer division. BCD arithmetic, Design of ALU. Concepts of instruction formats and instruction set, Instruction set types, types of operands and operations. Generation of memory address and addressing modes.	
Unit-III	**12 Lectures**
STACKS and QUEUS, GPR organization and stack based organizations. Encoding of machine instructions. Features of RISC and CISC processors. Processing unit design: Processor micro architecture –I Fundamentals concepts for data path implementation. Processor micro architecture-II data path implementation.	
Unit-IV	**12 Lectures**
Instruction pipelining, instruction pipelining hazards, data dependency hazards and control hazards. Overcoming hazards, instruction set design influence on pipelining. Parallel processing and pipelining, pipelining in RISC and CISC processors. Super scalar processors. In order and out of order execution.	
Unit-V	**12 Lectures**
Instruction level parallelism, introduction to VLIW processors, vector processors, CACHES: Data caches, instruction caches and unified caches Cache implementations. Fully associative and direct mapped caches. Write back versus write through caches. Input Output organization, accessing I/O devices, Interrupts. Memory mapped I/O and I/O mapped I/O.	

TEXT BOOK:

1. Computer Architecture: Scahaum's outlines by Dr. Rajkamal

REFERENCE BOOKS:

1. The 8088 and 8086 Microprocessors by Walter A. Tribel, Avtar Singh
2. Computer Organization & Architecture by William Stallings
3. Computer Architecture & Parallel Processing, Hwang & Briggs, McGraw Hill Microprocessor Archi. Prog. and app. With 8085/8080 by Ramesh S. Gaonkar.

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – T205 – CORE COURSE V –
COMPUTER ORIENTED NUMERICAL AND STATISTICAL METHOD

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Solution of non-linear & transcendental equations Computer Arithmetic: Floating-point representation of numbers, arithmetic operations with normalized floating-point numbers and their consequences, significant figures. Error in number representation-inherent error, truncation, absolute, relative, percentage and round-off error. Iterative Methods: Bisection method, method of false position, Newton Rapson method, Secant Method, Method of successive approximation.	
Unit-II	**10 Lectures**
Solution of linear equations Meaning, conditions for solutions, solution of equation by direct methods - (Gaussian elimination, Gaussian Jordan), iterative methods - (Jacobi method, Gaussian Seidel), ill-conditional equations and solution.	
Unit-III	**10 Lectures**
Interpolation and approximation Introduction, finite differences, Newton's formulae, Interpolation with unevenly spaced points, divided difference and their properties, inverse interpolation.	
Unit-IV	**10 Lectures**
Numerical integration & solution of ordinary differential equations Concept of numerical integration, trapezoidal method, Simpson - 1/3 rule, Simpson - 3/8 rule, understanding and solution of Ordinary Differential Equation and theoretical consideration, Euler method, modified Euler's method, R-K 4th order method, predictor corrector methods.	
Unit-V	**08 Lectures**
Statistics Graphical representation, Frequency distributions, Measures of central tendency, Measures of dispersions, Correlation, Regression.	

TEXT BOOK:

1. V. Rajaraman, Computer Oriented Numerical Methods, Prentice Hall, India

REFERENCE BOOKS:

1. S. S. Sastry, Introductory Methods of Numerical Analysis
2. M. K. Jain, S.R.K. Iyengar & R. K. Jain, Numerical Methods for Scientific and Engineering Computation
3. H. C. Saxena, Finite Differences and Numerical Analysis
4. Modes A., Numerical Analysis for Computer Science

Syllabus

M.Sc. (C.S.). Part I – Semester II

MCS – P206 – CORE COURSE VI

PRACTICAL COURSE IN PRACTICAL ON DATA STRUCTURE USING C++

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and mathematical under the guidance of respective faculty.

MCS – P207 – CORE COURSE VII

PRACTICAL COURSE IN PRACTICAL ON DATABASE MANAGEMENT SYSTEMS

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and mathematical under the guidance of respective faculty.

MCS – 208 (SKEG) – SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC) –

SKEG- T-103 – COMMUNICATIVE ENGLISH

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week: 03 Hours **Total Lectures: 48**

SKEG- T-119 – PERSONALITY DEVELOPMENT

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week: 03 Hours **Total Lectures: 48**

SKEG-T107 – FUNDAMENTAL OF BANKING & INSURANCE

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours **Total Lectures: 48**

SKEG-T108 – HEALTH EDUCATION

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours **Total Lectures: 48**

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – T301 – CORE COURSE I – OBJECT ORIENTED PROGRAMMING (USING JAVA)

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Introduction to Java: Features of Java, Object-oriented programming overview, Introduction of Java Technologies, How to write simple Java programs, Data Types, Variables, Memory concepts, decision making operators, Naming Conventions, Introduction to Class, Objects, Methods and Instance Variables, Primitive type Vs Reference Type, Initializing Objects with Constructors. Type conversion & casting, Operators, Control Statements(if Single-Selection Statement, if-else Double Selection), while Repetition Statement, for Repetition Statement, do-while Repetition Statements, switch Multiple-Selection Statement, break and continue Statements. Static Method, static field and Math Class, Activation Record, Casting, Scope of declaration and Method Overloading.	
Unit-II	**10 Lectures**
String Handling: The String constructors, String operators, Character Exaction, String comparison, String Buffer. Arrays: Declaring and Creating Arrays, Enhanced for Statement, Passing Arrays to Method, Multidimensional Arrays, Variable-Length Argument lists, Using Command-line Arguments. Final Instance Variables, this reference, static import, overloaded Constructors, Garbage collection and method finalize, Overloading methods, Parameter passing.	
Unit-III	**10 Lectures**
Inheritance: Extending classes protected Members, relationship between Superclasses and Subclasses, Using super, Constructor in Subclasses, The Object Class, Object Copying in Java. Polymorphism: Method overriding, upcasting, Dynamic Method Dispatch, final Method and classes, Abstract classes and Methods, instanceof operator, Downcasting, Class, Runtime type Identification. Packages and Interfaces: Defining a Package, Understanding CLASSPATH, Access Protection, Importing packages, Creating own packages. Defining an Interface, Properties of interface, advantages of interface Achieving multiple inheritances through interfaces, Variables in Interfaces, Comparable interface.	
Unit-IV	**10 Lectures**
Exception Handling: Introduction, overview of doing it and keywords used, when to use it, Java Exception Hierarchy, finally block, chained exceptions, declaring new exception types, preconditions and post conditions. Streams and Files: Introduction, Data Hierarchy, Files and Streams, Sequential-access Text Files, Object Serialization, Random-Access files, Java Stream class Hierarchy. Multithreading: What are threads, The java thread model, Thread priorities, Thread life cycle, Creating thread and executing thread, Thread Synchronization, producer-consumer problem without Synchronization.	
Unit-V	**08 Lectures**
Introduction To GUI : Introduction, Overview of swing Components, Displaying text and Images in a window, Introduction to Event Handling, Common GUI Event Type and Listener Interfaces, How Event Handling Works, Adapter Classes, Layout Managers. Applets: Applet basics, Applet Architecture, Applet life cycle methods, Applet HTML Tag and attributes, Executing applet in web browser and in the applet viewer, in Passing parameters to Applets, doing GUI programming in applet. Database connectivity: JDBC, The design of JDBC, Typical uses of JDBC, The Structured Query language, Basic JDBC Programming concepts, Executing Queries.	

BOOKS:

1. JAVA How to Program by Deitel & Deitel, Pearson Education, Sixth Edition
2. Java2 : The Complete Reference by Herbert Schildt, Tata McGraw- Hill, fourth Edition
3. Programming with Java (Schaum's Easy Outline) by John Hubbard
4. JAVA 2 Black Book
5. Thinking in Java by Bruce Eckel , Prentice Hall
6. Core Java 1.2: Volume 1 Fundamentals by Gary Cornell, Cay Horstmann, Prentice Hall
7. The Sun Microsystems Press Java Series
8. Java Servlet Programming, Janson Hunter with William Crawford, O'Reilly
9. JAVA How to Program by Deitel & Deitel, Pearson Education, Sixth Edition

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – T302 – CORE COURSE II – DATABASE APPLICATIONS AND TOOLS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**08 Lectures**
Database Environment: Data versus information, traditional file processing, disadvantages, database approach, range of database application, advantages of database approach. Components of database environment, evolution of database system.	
Unit-II	**12 Lectures**
Database Development Process: Information engineering, information architecture, planning, SDLC, CASE etc., Data Models: Relational, Hierarchical & Network Model with Examples. SDLCsteps, three-schema architecture for database development. **Modeling Data in the Organisation:** Modeling of the rules of organization, data names and definitions, ER model constructs entities and its types, attributes, relationships, degree, unary, binary, ternary, n-ary, cardinalities constraints, ER modeling examples.	
Unit-III	**12 Lectures**
Enhanced ER modeling: Supertype, subtypes, specialization, generalization, specifying constraints in EER models, Relational model properties, keys, primary, secondary, composite, properties of relations. Codd's rules, integrity constraints, creating relational tables, Transform EER diagrams into relations, seven different steps for mapping EER model into relations.	
Unit-IV	**08 Lectures**
Introduction to normalization, steps, functional dependencies, basic normal forms, definition of first, second, third normal form and removing anomalies from the relations. De-normalization and merging relations.	
Unit-V	**08 Lectures**
Special Topics (Overview) : Data Warehousing, Data Mining, Distributed Databases, Object oriented modeling, definitions, activities in phases of model development, advantages of OOM, UML class diagrams, Example of a model development.	

BOOKS:

1. "Modern Database Management" Seventh Edition, Hoffer, Prescott, McFadden Pearson Education
2. "Database Systems" Thomas M. Connolly, Carolyn E. Begg Pearson Education
3. Raghu R and Johannes G., "Database management Systems", Mc Hill 3rd Ed 2002,
4. Elmasri R, Navathe S, "Fundamentals of Database Systems", Addison Wesley 4th Ed.

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – T303 – CORE COURSE III – THEORY OF COMPUTATION

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
The Theory of Automata: String, Alphabets and Languages, Finite Automata, Finite State Machine, Basic Definition. Description of a Finite Automaton, Deterministic Finite Accepters- Transition Graphs, Languages, Non- Deterministic Finite Acceptors- Definition, Equivalence of Deterministic and Nondeterministic Finite Accepters, Mealy and Moore models- Definitions, Transformation of Mealy Machine into Moore Machine and vice-versa.	
Unit-II	**10 Lectures**
Conversion of NDFA to DFA, Finite Automata with ϵ transition, Removal of ϵ transition. The Myhill-Nerode theorem and Minimization of Finite Automata, Definition and Construction of Finite Automata. **Properties of Regular Sets:** Pumping lemma for regular set, Closure properties of regular set.	
Unit-III	**09 Lectures**
Formal Language: Basic Definition, Chomsky Classification of languages, Initialization of Finite Automata, Regular Expression, conversion of Regular Expressions into finite Automata and vice versa, Regular Grammars – Right and Left Linear Grammars, Equivalence between Regular Languages and Regular Grammars.	
Unit-IV	**09 Lectures**
Context-Free Grammars: Left most and Right most Derivations, Derivation Trees, Parsing and Ambiguity, Simplification of CFGs, Chomsky Normal Form, Greibach Normal Form, Cocke-Kasami-Younger Algorithm, Properties of Context-Free Languages.	
Unit-V	**10 Lectures**
Pushdown Automata: Definition, Non deterministic Pushdown Automata, Pushdown Automata for Context Free Languages. Context-Free Grammars for Pushdown Automata. Deterministic Pushdown Automata and Deterministic Context-Free Languages. **Turing Machine:** Definition of Standard Turing Machine, Construction of Turing Machine, Turing Machine as Language Accepters and Transducers.	

BOOKS:

1. Mishra and Chandrasekaran, Theory of Computer Science, Prentice Hall of India
2. Hopcraft and Ullman, Introduction to Automata Theory, Languages and Computation, Narosa Publishing House.
3. Moll, Arbib and Kfoury, An Introduction to Formal Language Theory, Springer-Verlag.
4. Peter Linz, An Introduction to Formal Languages and Automata, Narosa Publishing House.

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – T304 – CORE COURSE IV – COMPUTER GRAPHICS & MULTIMEDIA

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**08 Lectures**
Introduction: Introduction to Computer Graphics, Application of Graphics. **Display Devices:** Refresh Cathode -Ray Tubes, Raster Scan Displays, Random Scan Displays, Color CRT Monitors, Flat Panel Displays. Video cards/display cards **Input Devices:** Mouse, Trackball, Space ball, Data Glove, Joystick, Light pen, Scanner, Digital Camera, Touch Panels, Voice Systems. **Hardcopy Devices:** Printers and Plotters.	
Unit-II	**12 Lectures**
Graphics Primitives: **Line Generation Algorithms:** DDA algorithm, Bresenham's algorithm. **Circle Generation Algorithms:** Midpoint Circle algorithm, Bresenham's circle generation algorithm. **Ellipse Generation algorithm.** **Polygon filling Algorithms:** Scan Line Polygon fill algorithm, Inside - Outside Tests, Boundary-Fill algorithm, Flood -Fill algorithm.Fundamentals of aliasing and Ant aliasing Technique.	
Unit-III	**10 Lectures**
Clipping: Clipping operations, Point clipping, Text clipping, Exterior clipping. **Line clipping:** Cohen Sutherland Algorithm, Liang Barsky Algorithm, Nicholl-Lee-Nicholl Algorithm. **Polygon clipping:** Sutherland- Hodgeman Algorithm, Weiler Atherton Algorithm.	
Unit-IV	**10 Lectures**
Two Dimensional Transformations: Translation, Scaling, Rotation, Reflection, Shear, Homogenous coordinate system, composite transformations, raster method of transformation **Two Dimensional Viewing:** Window to View port coordinate transformation, **Three Dimensional:** 3D Geometry, 3D display techniques, transformations. **Projections:** Parallel Projection, Perspective Projection.	
Unit-V	**08 Lectures**
Color Models and Color Application: Color models: Properties of Light. Standard Primaries and the Chromaticity Diagram, XYZ Color Model, CIE Chromaticity Diagram. RGB Color Model, YIQ Color Model, CMY Color Model, HSV Color Model. Conversion between HSV and RGB Models. HLS Color Model, Color Selection and Application. **Advancements in the technology in Computer Graphics.**	

BOOKS:

1. Computer Graphics: Donald Hearn and M. Pauline Baker, Second Edition, Prentice Hall of India.

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – T305 – CORE COURSE V – COMPUTER NETWORKS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Introduction: Computer Network, Goals and Applications, Reference models – OSI and TCP/IP. A Comparative study. LAN, MAN and WAN and topologies, LAN components – File server, Workstations, Network Adapter Cards. Connection Oriented and Connection less services, Switching Techniques – Circuit Switching, Packet Switching.	
Unit-II	**10 Lectures**
Data Link Layer**:** Design Issues, Framing, Error Detection: Parity Check, Check Sum and Cyclic Redundancy Check (CRC); Correction Technique: Hamming code. Flow Control: Elementary Data Link Protocols: An Unrestricted Simplex Protocol, Simplex Stop-and-Wait Protocol, Sliding Window Protocols: One-Bit Sliding Window Protocol Go Back N and Selective Repeat. Data link layer in the Internet: SLIP and PPP.	
Unit-III	**09 Lectures**
MAC Sub-layer**:** Multiple access protocols: Aloha, CSMA Protocols; Collision- Free Protocols; IEEE MAC Sub-layer protocols: 802.3, 802.4, 802.5 and their management. High speed LANs – Fast Ethernet, FDDI, Wireless LANs	
Unit-IV	**09 Lectures**
Network Layer: Design issues, Routing Principles. Routing Algorithms: Optimality Principle, Shortest Path Routing, Flooding, Distance Vector Routing. Link State Routing, Hierarchical Routing, Broadcasting Routing, Multicast Routing, The Network Layer in the Internet: Internet Protocol, Internet addressing and Internet Control protocols.	
Unit-V	**10 Lectures**
Transport Layer: Elements of Transport Protocols, Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing. The Internet Transport Protocol UDP: Introduction to UDP. Introduction to TCP, The TCP Service Model, The TCP Protocol. Application layer**:** Client Server Architecture, DNS, WWW and HTTP, Cookies, Proxy Server. E-mail Protocols (SMTP, POP3, IMAP, MIME), FTP, TELNET. Network Security: Cryptography, Symmetric-key Algorithms, Public- key Algorithms, Digital Signatures.	

BOOKS:

1. Computer Networks, Andrew S. Tanenbaum, Addison-Wesley, 4th Ed
2. Data Communications and Networking, B.A. Frouzan, McGraw-Hill

Class Assignments:			
Assignment 1:			
	1.	Define the following:	
		i. Computer Networks	**ii.** Autonomous Computer
		iii. Distributed System Vs Computer Networks	**iv.** LAN, MAN, WAN
		v. Protocols, Interfaces and Services	**vi.** Internet
		vii. Host or End System	**viii.** Subnet
	2.	Write the need of layered architecture.	
	3.	Explain OSI and TCP/IP Model. Also write the development stages of these models(i.e. When these model came., who proposed and further development).	

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – T305 – CORE COURSE V – COMPUTER NETWORKS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

Class Assignments:					
Assignment 2:					
	1.	A bit string, 0111101111101111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?			
	2.	An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?			
	3.	What is the remainder obtained by dividing x7 + x5 + 1 by the generator polynomial x3 +1?			
	4.	What is the mechanism of sliding window flow control?			
Assignment 3:					
	1.	Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.			
	2.	Sketch the Manchester encoding for the bit stream: 0001110101.			
	3.	Describes Carrier Sense Multiple Access Protocols in short?			
	4.	Difference between slotted and pure aloha?			
Assignment 4:					
	1.	Give two example computer applications for which connection-oriented service is appropriate. Now give two examples for which connectionless service is best.			
	2.	Describes distance vector and link state routing algorithm?			
	3.	What are the functions of router, bridges, repeaters, and gateways?			
	4.	Difference between adaptive and non adaptive routing?			
	5.	Describes Dijkstra algorithm?			
Assignment 5:					
	1.	Explain different classes of in Internet address?			
	2.	Explain purpose of ARP, RARP, ICMP, and IGMP?			
	3.	What is host id and net id?			
	4.	What is the purpose of sub netting?			
	5.	Explain IP datagram header format?			
	6.	Explain UDP and TCP datagram header format?			
Assignment 6:					
	1.	Write in short about following:-			
		• TELNET	• FTP	• SMTP	• POP
		• MIME	• HTTP	• URL	• CGI
		• DNS			

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – P306 – CORE COURSE VI

PRACTICAL COURSE IN DATABASE APPLICATION

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 queries which are simple, advanced and mathematical under the guidance of respective faculty.

MCS – P307 – CORE COURSE VII

PRACTICAL COURSE IN OBJECT ORIENTED PROGRAMMING (USING JAVA)

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and mathematical under the guidance of respective faculty.

MCS – P308 – CORE COURSE VIII

PRACTICAL COURSE IN COMPUTER GRAPHICS

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and mathematical under the guidance of respective faculty.

Syllabus

M.Sc. (C.S.). Part II – Semester III

MCS – 308 (SKEG) – SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC) –

SKEG-T/P124 – PHP PROGRAMMING

MAX. MARKS: 70 + 30 — **MIN. PASS MARKS: 28 + 12**

PART A : THEORY - MAX. MARKS: 50 + 30 — **MIN. PASS MARKS: 20 + 12**

No. of Lectures per week : 02 Hours — **Total Lectures: 32**

PART B : PRACTICALS - MAX. MARKS: 20 — **MIN. PASS MARKS: 08**

No. of Laboratory per week: 02 Hours — **Total Lectures: 32**

SKEG- T-119 – PERSONALITY DEVELOPMENT

MAX. MARKS: 70 + 30 — **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week: 03 Hours — **Total Lectures: 48**

SKEG-T116– MANAGERIAL SKILLS

MAX. MARKS: 70 + 30 — **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours — **Total Lectures: 48**

SKEG-T108 – HEALTH EDUCATION

MAX. MARKS: 70 + 30 — **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours — **Total Lectures: 48**

Syllabus

M.Sc. (C.S.). Part II – Semester IV

MCS – T401 – CORE COURSE I – UNIX /LINUX ADMINISTRATION

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**08 Lectures**
Background: Evolution of Unix OS. Features of Unix operating system, Structure of File System. **Linux operating system:** Development of Linux. Applications of Linux operating system.	
Unit-II	**10 Lectures**
Basic UNIX Environment: Basic Commands, Directory management, pipes, tee, I/O redirection and other utilities. **Unix System Architecture and Process Management:** Shell, Shell Process, Parent and Child Process, Process Status, System Process, Background and Foreground Process, Termination of Process, Pattern matching, Navigating the File Systems.	
Unit-III	**10 Lectures**
Unix editor: VI editor, Creating new files. Text addition, deletion and changes. Dealing with sentences and paragraphs. Searching. Cut, paste and copy. **Shell programming**: Features of shell. Shell variables. Control statements. **Communication and Scheduling Process:** Bulletin board, message of the day, two way communication, insulation from the other users, mailbox, address all users, delay, execute at later running jobs periodically. .profile.	
Unit-IV	**10 Lectures**
Structure of Unix operating system: Structure of Unix kernel, Unix system calls. **Unix system**: File system calls, Process management calls. **Advance Filter**: Awk, Number processing, Interface with shell, functions**.**	
Unit-V	**10 Lectures**
Unix system administration: Adding and removing users. User accounting. Adding and removing hardware. Performing backups and restore. Disk space management. Introduction to client management in Unix and Linux. Role of System Administrator, Network management in Unix. Performance analysis. Unix Desktop.	

BOOKS:

1. Unix Operating Systems, Sumitabh Das, Tata McGraw Hills Publication
2. Unix System Administration Handbook (Second edition), Evi Nemeth, Garth Synder, Scott Seebass, Trent R Hein, Pearson Education - Asia, 2000
3. Design of UNIX Operating System, Maurice J. Back, Pearson Education - Asia

Syllabus

M.Sc. (C.S.). Part II – Semester IV

MCS – T402 – CORE COURSE II – COMPILER DESIGN

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**10 Lectures**
Compiler, Translator, Interpreter definition, Phase of compiler introduction to one pass & Multi pass compilers. Analysis of source program, duties of 6 phases. Review of Finite automata lexical analyser, Input, buffering, Recognition of tokens, Idea about LEX: A lexical analyser generator, Error handling.	
Unit-II	**10 Lectures**
Introduction to parsing. Bottom up parsing Top down parsing techniques. Shift reduce parsing, Operator precedence parsing, Recursive descent parsing predictive parsers. LL grammars & passers error handling of LL parser. LR parsers, Construction of SLR.	
Unit-III	**10 Lectures**
Conical LR & LALR parsing tables, parsing with ambiguous grammar, Syntax directed definitions; Construction of syntax trees, L~attributed definitions, Top down translation. Intermediate code forms using postfix notation and three address code.	
Unit-IV	**10 Lectures**
Representing TAC using triples and quadruples, Translation of assignment statement. Boolean expression and control structures. Definition of basic block control flow graphs, DAG representation of basic block.	
Unit-V	**08 Lectures**
Advantages of DAG, Sources of optimization, Loop optimization, Idea about global data flow analysis, Loop invariant computation, Peephole optimization Issues in design of code generator, A simple code generator, Code generation from DAG. Code Optimization.	

BOOKS:

1. Compiler Design, Aho, Ullman, Sethi
2. Compiler Construction Theory & Practice, Barrett, Bates, Gustafson, Couch
3. The Theory & Practice of Compiler Writing, Jean Paul Tremblay, Paul G. Sorenson
4. Recommended Books and Reference Material (Journals, Reports, etc) (Attach List)

Syllabus

M.Sc. (C.S.). Part II – Semester IV

MCS – T403 – CORE COURSE III – INTERNET AND WEB TECHNOLOGY

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 04 Hours **Total Lectures: 64**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**12 Lectures**
Introduction Dynamic Web Programming, HTML Forms, scripting languages, Introduction to HTTP, web Server and application notes Servers, Installation of Application servers, Configuration files, Web.xml. Java Servlet, Servlet Development Process, Deployment Descriptors, The Generic Servlet Lifecycle.	
Unit-II	**14 Lectures**
Servlet Packages, Classes, Interfaces, and Methods, Handling Forms with Servlets, Various methods of Session Handling. Various elements of deployment descriptors. Java Database Connectivity: various steps in process of connection to the database, Various type of JDBC Driver. JDBC objects properties and their connectivity with MySql.	
Unit-III	**14 Lectures**
Connection of JSP and Servlet with different database viz. Oracle, MS-SQL Server, MySQL. Java.sql Package. Accessing metadata from the database. Type of Statement, Connection pooling: multiple users and need of connection pooling.	
Unit-IV	**12 Lectures**
JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects. Writing JSPs. Expression Language (EL), Separating Business Logic and Presentation Logic, Building and using JavaBeans.	
Unit-V	**12 Lectures**
Session handling in JSP, Types of errors and exceptions handling, Standard Tab Library in JSP, Building Custom Tag Library, JSP Tag Library, MVC Design pattern, Advances in J2EE and other Web Technology.	

BOOKS:

1. Beginning Java EE 5, Kevin Mukhar, Chris Zelenak, James L Weaver
2. Core Servlets and Java Server Pages, 2nd edition, Marty Hall, Larry Brown, Pearson Education
3. Java Doc for various Technologies
4. Prefessional Java Server Programming, S. Allamaraju Wrox Press
5. Struts Recipes, G. Franciscus, Manning Press
6. Hibernate in Action, C. Bauer, G. King, Manning Press

Syllabus

M.Sc. (C.S.). Part II – Semester IV

MCS – T404 – CORE COURSE IV – DESIGN AND ANALYSIS OF ALGORITHMS

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per Week: 03 Hours **Total Lectures: 48**

The Question Paper will contain questions equally distributed in all Units. The Internal Choice will be given in all Questions.

Unit-I	**08 Lectures**
Order Analysis: Objectives of time analysis of algorithms; Big-oh and Theta notations. Master Theorem and its proof, solution of divide and conquer recurrence relations Searching, Sorting and Divide and Conquer Strategy; Linear Search, Binary Search,	
Unit-II	**12 Lectures**
Searching and Sorting: Divide and Conquer Strategy: Merge-sort; Quick-sort with average case analysis. Heaps and heap-sort. Lower bound on comparison-based sorting and Counting sort. **Dynamic Programming:** methodology and examples (Fibonacci numbers, Knapsack problem and some other generalized and simple examples) Longest integer subsequence, Longest common subsequence ,Weighted interval scheduling	
Unit-III	**10 Lectures**
Greedy Method: Methodology, examples (lecture Scheduling, process scheduling) and comparison with DP (more examples to come later in graph algorithms) : Knapsack problem and some other simple examples **Graph Algorithms:** Basics of graphs and their representations. BFS,DFS, Topological sorting.	
Unit-IV	**10 Lectures**
Minimum spanning trees (Kruskal and Prim's algorithms and brief discussions of disjoint set and Fibonacci heap data structures). Shortest Paths (Dijkstra, Bellman-Ford, Floyd-Warshall). Hard problems and approximation algorithms. Problem classes P, NP, NP-hard and NP-complete, deterministic and nondeterministic polynomial-time algorithms. Approximation algorithms for some NP-complete problems.	
Unit-V	**08 Lectures**
Backtracking, Branch and Bound technique, String Matching, Knave algorithm, KMP algorithm, Parallel Algorithms	

BOOKS:

1. Algorithms, Cormen, Leiserson, and Rivest, MIT Press 2001
2. ALGORITHMS IN C++ by Robert Sedgewick, Pearson Education,,2008
3. Fundamentals of Computer Algorithms by Ellis Horowitz and Sartaj Sahni, Galgotia Publication 1998
4. Computer Algorithms Introduction to Design & Analysis by Sara Baase and Allen Van Gelder, Pearson Education 1999

ELECTRONIC MATERIALS, WEB SITES ETC

http://www.acm.org/, http://www.ieeexplore.ieee.org/Xplore/dynhome.jsp

Syllabus

M.Sc. (C.S.). Part II – Semester IV

MCS – P405 – CORE COURSE V

PRACTICAL COURSE IN PRACTICAL ON UNIX /LINUX ADMINISTRATION

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

Installation Process of UNIX and LINUX.

The Students have to prepare a list of minimum 50 commands and 10 shell programs which are simple and advanced under the guidance of respective faculty.

MCS – P406 – CORE COURSE VI

PRACTICAL COURSE IN PRACTICAL ON INTERNET AND WEB TECHNOLOGY

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 programs which are simple, advanced and Web oriented under the guidance of respective faculty.

MCS – P407 – CORE COURSE VII

PRACTICAL COURSE IN PRACTICAL ON DESIGN AND ANALYSIS OF ALGORITHMS

MAX.MARKS: 50 **MIN. PASSING MARKS: 20**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to prepare a list of minimum 50 algorithms / programs which are simple, advanced and mathematical under the guidance of respective faculty.

Syllabus

M.Sc. (C.S.). Part II – Semester IV

MCS – 408 (SKEG) – SKILL ENHANCEMENT / GENERIC COURSE - ANY ONE (SEC / GC) –

SKEG-T/P117 – .NET PROGRAMMING (DOT NET PROGRAMMING) (USING C#)

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

PART A : THEORY - MAX. MARKS: 50 + 30 **MIN. PASS MARKS: 20 + 12**

No. of Lectures per week : 02 Hours **Total Lectures: 32**

PART B : PRACTICALS - MAX. MARKS: 20 **MIN. PASS MARKS: 08**

No. of Laboratory per week: 02 Hours **Total Lectures: 32**

SKEG- T-119 – PERSONALITY DEVELOPMENT

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week: 03 Hours **Total Lectures: 48**

SKEG-T107 – FUNDAMENTAL OF BANKING & INSURANCE

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours **Total Lectures: 48**

SKEG-T108 – HEALTH EDUCATION

MAX. MARKS: 70 + 30 **MIN. PASS MARKS: 28 + 12**

No. of Lectures per week : 03 Hours **Total Lectures: 48**

MCS – P409 – PROJECT / INTERNSHIP

MAX.MARKS: 100 **MIN. PASSING MARKS: 40**

No. of Laboratory per Week: 04 Hours **Total Lectures: 64**

The Students have to complete a Project or Internship under the guidance of respective faculty.
