

**UNIVERSITY OF
MUMBAI**

**Revised syllabus for
F.Y.BSc. Computer Science
to be implemented**

from

2008-2009

Proposed syllabus and examination scheme
for
F.Y.B.Sc. Computer Science (Theory and Practical)
from
Academic year 2008-09
Conduct of Course

The workload per paper for theory and practical workload per batch, batch size for practical and passing standards will be as per the existing norms / regulations and University Guidelines.

Examination Scheme

Theory (Examination duration per paper 2 hours)				
	Title	Examination	Maximum Marks	Maximum Marks after conversion
Paper I	Computer organization:	First Term	60	30
	Introduction to microprocessors and computer architecture	Second Term	60	30
Paper II	Algorithms and Programming in C	First Term	60	30
		Second Term	60	30
Practical (Examination duration per paper 3 hours)				
Paper I	Computer organization: Introduction to microprocessors and computer architecture	Annual Actual Practical	30	30
Paper II	Algorithms and Programming in C	Annual Actual Practical	30	30
Certified Journal	Paper I and Paper II	Annual	10	10
Viva-voce		Annual	10	10
Evaluation Scheme (For Paper I & Paper II) (As per the recommendations passed by the Science Faculty)				
Examination	Maximum Marks		Maximum marks after conversion	
	Paper I	Paper II	Paper I	Paper II
First Term	60	60	30	30
Second Term	60	60	30	30
Practical	40	40	40	40
Total			100	100

Paper Pattern (For Paper I and Paper II)
(All questions are compulsory)

Questions	Term I	Term II	Maximum Marks*
Q1.	Based on Unit 1, 2 & 3	Based on Unit 4, 5 & 6	15
Q2	Based on Unit 1	Based on Unit 4	15
Q3	Based on Unit 2	Based on Unit 5	15
Q4	Based on Unit 3	Based on Unit 6	15
	Total		60

*In each question; maximum marks with options should be set for 22 or 23 marks with internal options.

General Guidelines

- Each paper is divided into six equal units. **First 3 units** of each paper are expected to be taught during **first term** and **next 3 units** are expected to be taught during **second term**. The lectures allocated for the respected units are suggestive.
- Minimum 75 % experiments from each paper are required to be completed and written in the journal.

Syllabus
Paper I
Computer organization
Introduction to microprocessors and computer architecture
Theory
Section I

Unit I: (15 Lectures)

Introduction :

Computers: History of computers and their classification

Basics of modern computer systems: View of a computer as an integrated system, Neumann machine, block diagram of a computer system.

Information - Definition, Characteristics and interpretation, Data and its logical and physical concepts, binary form of program and instruction.

Number Systems – Binary, Decimal, Octal, Hexadecimal and their inter-conversions.

Computer Arithmetic – Binary addition and subtraction using signed-Magnitude, 1's complement and 2's complement, Binary multiplication and division, Floating point representation and arithmetic, arithmetic through stacks.

Codes for character representation – hexadecimal, BCD, Excess-3, Gray code, ASCII, EBCDIC, Unicode.

Ref: 1) Computer organization and architecture:

William Stallings, PHI, Sixth edition

2) Computer System architecture:

M. Morris Mano, PHI, edition

Unit II: (15 lectures)

Digital logic circuits:

Boolean algebra – Basic identities of Boolean Algebra, Boolean function

Logic Gates – AND, OR, NOT, NOR, NAND, EX-OR EX-NOR operations and their truth table, Minimization of gates by K-maps.

Digital Circuits – Half Adder, Full Adder, Binary adder-subtractor, binary incrementer, Multiplexers, Encoder and decoder

Flip Flops – Concept of sequential circuits, concept of clock and synchronization, S-R, J-K, Preset and Clear, Master-Slave J-K, D, T Flip Flops, their truth tables and identities, Conversion from one type to another type of Flip Flop, concept of counters and registers, shift registers.

Ref: 1) Computer System architecture: M. Morris Mano

PHI, edition

Unit III: (15 lectures)

Introduction to computer components:

Memory –

Primary Memory – RAM, SRAM, DRAM, ROM, EPROM.

Secondary Memory – Magnetic Floppy and Hard Disk.

Optical Memory – CDROM, WORM.

Concept of Virtual Memory

Concept of Cache and their need.

Memory hierarchy.

Input/output devices –

Input/output devices, input/output interface, asynchronous data transfer, modes of data transfer..

CPU –

Functions of CPU, register classification and organization, instruction sets and examples of instruction set, addressing schemes, instruction formats, instruction cycle and instruction pipelining.

- Ref:** 1) Computer organization and architecture:
William Stallings, PHI, Sixth edition
2) Computer System architecture:
M. Morris Mano, PHI, edition

Section II

Unit IV: (15 lectures)

Internal memory organization:

DRAM, SRAM, ROM types, Cache Memory Principles, elements of cache design, Pentium 4 cache.

External Memory organization:

Magnetic disk, RAID, Optical memory, Magnetic tape

Input/Output device organization:

External devices, I/O modules, Concepts of programmed I/O, interrupt Drive I/O, DMA, I/O processors.

- References: 1) Computer organization and architecture:
William Stallings, PHI, Sixth edition
2) Computer System architecture:
M. Morris Mano, PHI, edition

Unit V: (15 lectures)

Operating System Support:

Basic Concepts, Batch, Multiprogramming and Time-Sharing, scheduling, scheduling, Memory Management.

Introduction to multiprocessors:

Characteristics of Multiprocessors, Time-Shared Bus, Multi-port memory.

Unit VI: (15 lectures)

Introduction and programming with Microprocessors:

Introduction to 8085 Architecture and its extension to architecture to 8086, functional block diagram, Bus interface unit, Execution unit, general purpose registers, segment registers, pointers and index registers basic instruction set and organization of 8086,. 8086 Assembly language programming.

References:

1. Microprocessor architecture, programming and applications with 8085:
Ramesh Gaonkar, Fourth edition, Penram international.
2. Introduction to 8086 programming:

Practical Section I

- A) Introduction to Operating system desktop, folders, files, shortcuts, popular menus, using notepad, word, excel, power point.
- B) Introduction to windows wildcard characters, absolute path, relative path and commands like md, cd, rd, copy, ren, del etc.

- 1) Demo practical on various internal and external parts of computer and their interconnection/working.
- 2) Demo hands on assembly of PC.
- 3) Study of basic gates.

- 4) Implementation of Boolean equations using basic gates.
- 5) Study of flip-flops.
- 6) Study of 4 to 1 multiplexer.
- 7) Study of decoder
- 8) Study of counters.
- 9) Study of universal shift registers.
- 10) Study of 4 bit adder/subtractor.

Note: 1) Practical A and B are compulsory. They are to be written in journal but should not be the part of practical examination.

2) **Eight** practical from the list should be performed.

Section II

- 1) Demo practical on working of 8085.
- 2) Study of internal memory, I/O modules.
- 3) Study of operating system.
- 4) Study of networking of computers and other devices.
- 5) Study of concepts of parallel processing.
- 6) Study of 8086 architecture.
- 7) Study of 8086 instruction set
- 8) Writing programs with 8086 microprocessor for
 - a) Addition of 1 to n numbers
 - b) Finding largest/smallest from n given numbers.
- 9) Writing program with 8086 microprocessor for demonstration of use of JUMP instructions.

- 10) Writing programs with 8086 microprocessor for
 - a) Use of I/O ports.
 - b) Block transfer of memory.

Note: 1) **Any five** experiments from 1 to 7 practical should be performed.

2) Experiments from 8 to 10 are compulsory.

Paper II Algorithms and Programming in C Theory Section I

UNIT I: INTRODUCTION TO ALGORITHMS AND C

(15 Lectures)

Fundamentals of algorithms: Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures.

Algorithmic problems: Develop fundamental algorithms for (i) Exchange the values of two variables with and without temporary variable, (ii) Counting positive numbers from a set of integers, (iii) Summation of set of numbers, (iv) Reversing the digits of an integer, (v) Find smallest positive divisor of an integer other than 1, (vi) Find G.C.D. and L.C.M. of two as well as three positive integers, (vii) Generating prime numbers.

Analysis of algorithms: Running time of an algorithm, worst and average case analysis.

Different approaches in programming: Procedural approach, Object Oriented approach, Event Driven approach.

Structure of C: Header and body, Use of comments, Compilation of a program.

Data Concepts: Variables, Constants, data types like: int, float char, double and void.

Qualifiers: short and long size qualifiers, signed and unsigned qualifiers. Declaring variables. Scope of the variables according to block. Hierarchy of data types.

UNIT II: BASICS OF C

(15 Lectures)

Types of operators: Arithmetic, Relational, Logical, Compound Assignment, Increment and decrement, Conditional or ternary, Bitwise and Comma operators. Precedence and order of evaluation. Statements and Expressions.

Type conversions: Automatic and Explicit type conversion.

Data Input and Output functions: Formatted I/O: printf(), scanf(). Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().

Iterations: Control statements for decision making: **(i)** Branching: if statement, else.. if statement, switch statement. **(ii)** Looping: while loop, do.. while, for loop. **(iii)** Jump statements: break, continue and goto.

UNIT III: ARRAYS, STRINGS AND SORTING TECHNIQUES

(15 Lectures)

Arrays: (One and multidimensional), declaring array variables, initialization of arrays, accessing array elements.

Strings: Declaring and initializing String variables. Character and string handling functions.

Sorting Algorithms: Bubble, Selection, Insertion and Merge sort, Efficiency of algorithms, Implement using C.

Section II

UNIT IV: FUNCTIONS, STRUCTURES, RECURSION AND UNION (15 Lectures)

Functions: Global and local variables, Function definition, return statement, Calling a function by value, Macros in C, Difference between functions and macros.

Storage classes: Automatic variables, External variables, Static variables, Register variables.

Recursion: Definition, Recursion functions algorithms for factorial, Fibonacci sequence, Tower of Hanoi. Implement using C.

Structure: Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures, structures and functions

Unions: Defining and working with union.

UNIT V: POINTERS AND FILE HANDLING

(15 Lectures)

Pointer: Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Chain of pointers, Pointers and Arrays, Pointers and Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers, Pointer to function, Pointer to structure, Pointers within structure.

File handling: Different types of files like text and binary, Different types of functions: fopen(), fclose(), fgetc(), fputc(), fgets(), fputs(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek()

Dynamic Memory Allocation: malloc(), calloc(), realloc(), free() and sizeof operator,

UNIT VI: LINK LISTS, STACKS AND QUEUES

(15 Lectures)

Linear Link lists: Representation of link list in memory, Algorithms for traversing a link list, searching a particular node in a link list, insertion into link list (insertion at the beginning of a node, insertion after a given node), deletion from a link list. Implement using C.

Stacks: Definition, Array representation of stacks, Algorithms for basic operators to add and delete an element from the stack, Implement using C.

Queues: Representation of queue, Algorithm for insertion and deletion of an element in a queue, Implement using C.

Recommended Books:

- (1) Introduction to Algorithms (Second Edition): *Cormen, Leiserson, Rivest, Stein, PHI(Chapter 1,2,3,10)*
- (2) Data Structures (Schaum's outline series in computers) : Seymour Lipschutz McGraw-Hill book Company (Chapter 2, 5, 6, 9)
- (3) Programming in ANSI C(Third Edition) : *E Balagurusamy TMH (Chapters: 2, 3, ,4, 5, 6, 7, 8, 9, 10, 11, 12, 13)*

Additional References:

- (1) Fundamental Algorithms (Art of Computer Programming Vol 1): Knuth Narosa Publishing House
- (2) Mastering Algorithms with C, Kyle Loudon, Shroff Publishers
- (3) Algorithms in C (Third Edition): Robert Sedgewick , Pearson Education Asia
- (4) Data Structures A Pseudocode Approach with C: Richard F. Gilberg, Behrouz A. Forouzan , Thomson
- (5) Let us C by Yashwant Kanetkar, BPB
- (6) Programming in ANSI C by Ram Kumar, Rakesh Agrawal, TMH
- (7) Programming with C (Second Edition):Byron S Gottfried (Adapted by Jitender Kumar Chhabra) *Schaum's Outlines (TMH)*
- (8) Programming with C : K R Venugopal, Sudeep R Prasad TMH Outline Series
- (9) Unix and C : M.P. Bhave and S.A. Pateker Nandu printers and publishers private limited

PRACTICAL

Suggestions while writing programs:

- 1) Use of comments at appropriate places is necessary.
- 2) Use appropriate indentation while nesting the loops, if-else statements.
- 3) Do not ignore warnings after the compilation.
- 4) Optimize the codes as far as possible, by using optimization techniques.

Section I

1. Convert the following algorithms using C:
Exchange the values of two variables with and without temporary variable.
2. Convert the following algorithms using C:
Counting positive numbers from a set of integers.
3. Convert the following algorithms using C:
Summation of set of numbers.
4. Convert the following algorithms using C :
Reversing the digits of an integer.
5. Convert the following algorithms using C:
Find smallest positive divisor of an integer other than 1.
6. Convert the following algorithms using C:
Find G.C.D. and L.C.M. of two as well as three positive integers.
7. Convert the following algorithms using C:
Generating prime numbers.

8. Write a program to find the
 - (a) sum of two matrices of order $m \times n$ and transpose of order $m \times n$ where $m, n \leq 3$.
 - (b) multiplication of two matrices of order m , where $m \leq 3$, finding square and cube of a square matrix.
 - (c) Inverse of a matrix
 - (d) $|A|$, $|B|$ and verify the identity $|A B| = |B A| = |A| \cdot |B|$, where $||$ denote determinant of the matrix and A and B of size 2×2 .
9. Write a program to (a) input a sentence (b) count the number of occurrences of the given pattern of letters (for instance 'est' or 'ed') (c) find the position of the rightmost or leftmost character occurred from the pattern of letters
10. Write a program which counts the number of (a) paragraphs occurred.(b) times the word "the" appears in a short story
11. Given an array S of n integers. Write a program to (a) sort the elements in S in ascending order by considering an array of n-2 elements by any sorting method, (b) find the median of elements of S

Section II

1. Write a program to create functions (a) to generate twin primes in a given range of numbers, (b) to find the prime factors of a given integer.
2. Write a program to accept details of 5 customers that includes customer number, name and mobile number. Create a menu with options 'Modify', 'Display' and 'Exit'. Write functions modify(), which will allow modification of mobile number and function display(), which will display all the details of customers.
3. Write a program to create structure to (a) find and print the average marks of five subjects along with the name of student. (b) store names of the states (within India) and their capital cities. Show the capital by inserting state from the keyboard.
4. Write a program using pointer notation (a) to write function to exchange two strings, (b) to determine whether the given string is a palindrome, (c) to find the average of each students in 3 tests. Number of students can be given from keyboard.
5. Write a program that accepts a number from the user and passes a pointer to the number to a function for processing. This function passes a pointer to this pointer to another function for processing of the number. Both the functions should display the number.
6. Write a function called increment that accepts a date structure with three fields. The first field contains the month (a pointer to string). The second field is an integer showing the day in month. The third field is an integer showing the year. The function should increments the date by 1 day and returns the new date. If the date is the last date in the month, the month field must be changed. If the month is December, the value of year must be changed when day is 31. A year is leap year if
 - i. It is evenly divisible by 4 but not with 100 or
 - b. It is evenly divisible by 400
7. Write a program to crate a dynamic one and two dimension array by accepting number of rows and/or columns from the user at runtime using pointer notation.
8. Write a program to (a) read string from the user to check whether it exists in a given file or not. (b)to accept a file name and then accept contents that should be stored in the file until the user types "end", (c) declare a structure representing student, accept data of 5 students and store it in a file, (d) convert a capital case letter file to small case letter file
9. Write a program to compute factorial of a large number.
10. The Ackerman's function $Ack(m, n)$ is defined recursively by
 - a. If $m = 0$ then $Ack(m, n) = n+1$,
 - b. If $m \neq 0$ but $n = 0$ then $Ack(m, n) = Ack(m-1, 1)$,
 - c. If $m \neq 0$ and $n \neq 0$ then $Ack(m, n) = Ack(m-1, Ack(m, n-1))$
 - d. Write a program to calculate $Ack(0, n)$, $Ack(4, 0)$, $Ack(6, 10)$
11. Fibonacci sequence is defined by
 - a. If $n = 0$ or $n = 1$ then $F_n = n$;
 - b. If $n > 1$ then $F_n = F_{n-2} + F_{n-1}$
 - c. Write a program to generate Fibonacci sequence.
12. Write a program to solve Tower of Hanoi problem

- 13.** Write a program to create a linked list containing student's name and marks (a) search marks of a particular student (b) insert a new student at the beginning of the list. (c) insert a new student at the end of the list (d) delete a particular student from a list.
- 14.** Write a program to create following stack of characters, where STACK is allocated N=8 memory cells:
1. STACK: A, C, D, F, K, ____, ____, ____.
 - ii. (For notation convenience, we use “__” to denote an empty memory cell). The program should describe the STACK after the following operations take place:

b. (a) POP (STACK, ITEM)	(e) POP (STACK, ITEM)
c. (b) POP(STACK, ITEM)	(f) PUSH(STACK, R)
d. (c) PUSH(STACK, L)	(g) PUSH(STACK, S)
e. (d) PUSH(STACK, P)	(h) POP(STACK, ITEM)
 - f. b The POP always deletes the top element from the stack, and the PUSH always adds the new element to the top of the stack.
- 15.** Write a program to create following queue of cities, where QUEUE is allocated 6 memory cells:
1. QUEUE: ____, Latur, Bashirabad, Rawalgav, Palanpur, ____
- (For notation convenience, we use “__” to denote an empty memory cell).
 Where FRONT = 2, REAR= 5. The program should describe the QUEUE, including FRONT and REAR, after the following operations take place:
 (a) Allahabad is added, (b) two cities are deleted, (c) Mumbai is added,
 (d) Mathura is added, (e) three cities are deleted and (f) Nagpur is added.