

Rs - 1 0 - -

॥ अतरी पेटवू ज्ञानज्योत ॥

10



North Maharashtra University,  
Jalgaon.

Syllabus for S.Y.B.Sc.

**COMPUTER SCIENCE**

**W.E. From June, 2003**

॥ संतती येदव् इलनव्दोर ॥  
**NORTH MAHARASHTRA UNIVERSITY, JALGAON**  
**Syllabus for S.Y.B.Sc. (Computer Science)**  
**(W.E.F. June, 2003)**

**Paper – I : Section – I**  
**Data and File Structures-I**

- |    |   |                  |
|----|---|------------------|
| 1. | <b>Introduction to file :</b>   | <b>L-12 M-12</b> |
|    | 1.1 Concept of field, Record and file.  |                  |
|    | 1.2 File structure  |                  |
|    | Definition and concept  |                  |
|    | Record organization   |                  |
|    | Sequential files  |                  |
|    | The structure of sequential file  |                  |
|    | Processing sequential file  |                  |
|    | Indexed sequential file   |                  |
|    | The structure of I.S.F.   |                  |
|    | Processing I.S.F.   |                  |
|    | Directed Files  |                  |
|    | The structure of direct file  |                  |
| 2. | <b>Introduction to Data Structures :</b>  | <b>L-12 M-10</b> |
|    | 2.1 Introduction to data structure with primitive and nonprimitive data structures.   |                  |
|    | 2.2 Algorithmic Notations   |                  |
|    | Format conventions, statements and control structures, data structures, Arithmetic operators and expressions, strings & string operators, relations & relational operators, logical operations & expressions, input & Output, subalgorithm. |                  |
|    | Algorithmic problem based on array insertion and merging.   |                  |
| 3. | <b>Linear data structures and their sequential storage representation:</b>  | <b>L-12 M-12</b> |
|    | 3.1 Storage structure of array  |                  |
|    | 3.2 Stacks  |                  |
|    | Definition and Concepts   |                  |
|    | Operations on stacks  |                  |
|    | 3.3 Application of stacks   |                  |
|    | Recursion   |                  |
|    | Polish expressions  |                  |
|    | Infix, Postfix and Prefix notation  |                  |
|    | Infix to Postfix  |                  |
|    | 3.4 Queues  |                  |
|    | 3.5 Priority Queues   |                  |
| 4. | <b>Linear data structures and their linked storage representation :</b>   | <b>L-16 M-16</b> |
|    | 4.1 Pointers and their linked allocation  |                  |
|    | 4.2 Linked linear list  |                  |
|    | Operations on singly linked storage structures  |                  |
|    | Circularly linked linear list   |                  |
|    | Doubly linked linear list   |                  |
|    | 4.3 Application of linked linear list   |                  |
|    | Polynomial manipulation   |                  |

**Paper – I : Section – II**  
**Data and File Structures-II**

- |    |   |                  |
|----|---|------------------|
| 5. | <b>Non linear structures:</b>                               | <b>L-10 M-10</b> |
|    | 5.1 Matrix representation of graphs                         |                  |
|    | 5.2 list structure  |                  |
|    | 5.3 Other representation of graphs                          |                  |
|    | 5.4 Breadth First Search                                    |                  |
|    | 5.5 Depth First Search                                      |                  |
|    | 5.6 Dijkstra's shortest path algorithm                      |                  |
| 6. | <b>Trees:</b>   | <b>L-20 M-16</b> |
|    | 6.1 Definition and Concepts                                 |                  |
|    | 6.2 Operations on binary trees                              |                  |
|    | 6.3 Storage representation and manipulation of binary trees |                  |
|    | Linked storage representation of B.T.                       |                  |
|    | Threaded storage representation of B.T.                     |                  |
|    | 6.4 Sequential and other representations of trees           |                  |
|    | 6.5 Search trees  |                  |
|    | Height balanced trees                                       |                  |
|    | Weight balanced tree  |                  |

7.	<b>Sorting and searching:</b>	L-12	M-16
	7.1	Sorting	
		Notations and concepts	
		Selection sort	
		Insertion sort	
		Bubble sort	
		Merge sort	
		Heap sort	
		Radix sort	
		Quick sort	
	7.2	Searching	
		Sequential searching	
		Binary searching	

<b>Chapter-8:</b>	<b>Hash table methods:</b>	L-10	M-8
	8.1	Introduction	
	8.2	hashing functions	
	8.3	Collision resolution techniques	

**Note:** Subject should be taught independent of any language

**Reference book:**

1. An introduction to data structures with application:  
Jeen Paul Tremblay, G. Sorenson  
Tata McGraw Hill, Edition-II

**Paper- II: Section -I  
Programming in C++**

- |    |  |       |       |
|----|--|-------|-------|
| 1. | <b>Elements of C++ :-</b>  | L- 6  | M- 4  |
|    | Comments, Header file, Input output streams, Manipulators, Additional datatypes and Operators in C++, Structures and unions in C++.  |       |       |
|    | Function :- Reference Arguments, inline function, default argument, variables and storage classes.   |       |       |
| 2. | <b>Objects and classes :-</b>  | L- 8  | M- 8  |
|    | Simple Classes (class specified, c++ objects, accessing class members, memory allocation to object, static class data) , Constructors and destructors, constant member functions, passing arguments as objects and returning.  |       |       |
| 3. | <b>Function Overloading :-</b>   | L- 8  | M- 8  |
|    | Overloaded function, overloaded operator, overloading unary and binary operators, multiple overloading, comparison operators, arithmetic, assignment operators, data conversion between objects and basic types.   |       |       |
| 4. | <b>Inheritance:-</b>   | L- 10 | M- 8  |
|    | Derived class and Base class, derived class constructors, class hierarchies, public and private inheritance, multiple inheritance, containment classes within classes, inheritance and program development, pointer to object, difference between pointers and references. |       |       |
| 5. | <b>Virtual functions and other substitutes :-</b>  | L- 10 | M- 8  |
|    | Virtual functions, pure virtual functions, friend functions, static functions, assignments and copy initialization, the copy constructors, This pointer, Abstract Classes.   |       |       |
| 6. | <b>Introduction to Templates :-</b>  | L- 5  | M- 4  |
|    | Introduction to templates and exception handling, function with templates.   |       |       |
| 7. | <b>Files and Streams :-</b>  | L- 5  | M- 10 |
|    | Streams, string input output, file pointer, Error handling, redirection, command line arguments, pointer output, overloading the << and >> operators.  |       |       |

**Reference :**

- 1) Turbo C++ programming -Robert Lafore.
- 2) Mastering in C++ - K. R. Venugopal.
- 3) C++ programming - E. Balajiswamy

**Paper- II : Section-II**  
**Database management systems (DBMS)**

1.	<b>Introduction</b>	L- 5    M-6
	1.1    Purpose of database system	
	1.2    View of data	
	1.3    Data model.	
2.	<b>Entity Relational model</b>	L- 6    M- 8
	2.1    Basic concept	
	2.2    Design Issue	
	2.3    Mapping cardinality	
	2.4    Keys	
	2.4.1    Primary key	
	2.4.2    Secondary key	
	2.4.3    Super key	
	2.4.4    Candidate key	
	2.4.5    Foreign key	
	2.5    E-R diagram	
	2.6    Strong and Weak entity set.	
3.	<b>Relational Model</b>	L-16    M-10
	3.1    Structure of relational database	
	3.2    Relational algebra	
	Select, Project, union, set difference, Cartesian product, rename, set intersection, natural join	
	3.3    SQL	
	Basic structure, select, where, from, rename, tuple variables, string operations, order by, union, intersect, except, group by, aggregate functions, null values, nested subqueries, views, modifications of the database, data definition language.	
4.	<b>Integrity Constraints</b>	L- 8    M-8
	4.1    domain constraints	
	4.2    referential integrity	
	4.3    Assertion and Triggers	
	4.4    Functional dependencies	
5.	<b>Relational database design</b>	L- 12    M- 10
	5.1    What is normalization	
	5.2    Types of normal forms	
	1NF, 2NF, 3NF, BCNF	
	(Examples are expected to cover)	
6.	<b>Concurrency control</b>	L- 3    M- 4
	6.1    Lock based protocol	
	Two phase locking protocol.	
7.	<b>Recovery system</b>	L- 4    M- 4
	7.1    Failure classification and storage structure	
	7.2    Log based recovery	
	7.3    Shadow paging	

**References:-**

1.    Database system concepts (THM) (1V<sup>th</sup> edition)  
        BY Henry F. Korth.
2.    Database management system principals.  
        BY C. J. Date.
3.    Database concepts and organization.  
        By James Martin.

4

**Paper-III : Section -I**  
**Data and File Structures**

1. Write a program using C to demonstrate Index sequential mechanism using files
2. Write a program using C for the following using stack,  
3. Basic operations of stack- push, pop
4. Write a program to convert infix to postfix expression
5. Write a program using C to circular queue operation insertion, deletion
6. Write a program using C to implement linked list that covers insert, delete operation of singly linked list

**C++ Programming**

1. Do a object oriented analysis of any system. Prototype all possible objects detected and represent the system using concepts of oops.
2. Write a C++ program using structure to implement string operations as structure members.
3. Write a C++ program using class described in 2).
4. Write a C++ program to demonstrate function overloading or operator overloading.
5. Write a C++ program to implement inheritance.
6. Write a C++ program to implement virtual function.
7. Write a C++ program to demonstrate use of template.
8. Write A C++ program to implement file handling using binary file operating functions.

**Section -II**  
**Data and File Structure**

1. Write a program using C to build binary search tree and perform inorder, preorder and postorder traversal of it
2. Write a program using C to implement BFS, DFS algorithm for graph
3. Write a program using C to implement Dijkstra's shortest path algorithm
4. Write a program using C to demonstrate use of hash table, collision resolution technique should be covered
5. Write a program using C for,  
6. merge sort  
7. quick sort  
8. radix sort
9. Write an object oriented program using C++ to reverse singly linked list using stack

X=X=X

C:\Syboccs.doc